Personalized Query Result Presentation and Offer Composition for E-Procurement Applications

Doctoral Thesis

Dipl.-Math. oec. Stefan Fischer

Faculty of Applied Computer Science
University of Augsburg

Fischer@Informatik.Uni-Augsburg.DE

1. Examiner: Prof. Dr. Werner Kießling
2. Examiner: Prof. Dr. Elisabeth André

Day of oral examination: 27.07.2004

Abstract

As long as there have been database search engines there has been the problem of what to present to the customer when there is no perfect match and how to present that query result to the customer. Respecting the customer's search preferences is the suitable way to search for best matching alternatives. Modeling such preferences as strict partial orders in "A is better than B" semantics has been proven to be user intuitive in various internet applications. The better the search result, the better is the psychological advantage of the presenter. Thus, there is the necessity to know the quality of the search result with respect to the search preferences. Moreover, for an e-procurement portal it is necessary not only to personalize the composition of the shopping cart but also the price determination for an offer.

This work introduces a novel personalized and situated quality valuation for query results. Based on a human comprehensible linguistic model of five quality categories a very intuitive framework for valuations is defined for numerical as well as for categorical search preferences. These quality valuations provide human comprehensible presentation arguments. Moreover, they are used to compute the situated overall quality of a search result. Then a flexible and situated filter decides which results to present, e.g. by respecting quality requirements of the customer. A so called presentation preference determines which results are predestined to be especially pointed out to the customer. This unique framework, realized as the Preference Presenter technology for query result presentation, enables a search engine to proactively present search results by respecting an underlying strategy, e.g. a special sales strategy.

For the first time it is possible to build a personalized and situated e-procurement portal. Preference based components are combined to effectively manage the work of a human sales agent via internet application. For the modeling of a personalized automatic offer composition widespread IT product standards like BMEcat and eCl@ss are exploited. Two new and extensible e-commerce components of flexible usage are designed, namely an electronic bargainer that is able to use techniques like up, cross, and down selling, and a personalized price offer including a flexible discount framework.

With COSIMA^{B2B} a use case is realized. In the evaluation it is shown that the duties of a human sales agent can be automated. Furthermore, experiments have shown that test customers react similarly to sales strategies that are applied by a computer instead of a human.

Moreover, the Preference Presenter enables lots of further e-commerce applications or advanced search engines to present their search results proactively and more comprehensibly.

Acknowledgments

At the chair for databases and information systems of the University of Augsburg (Germany) my doctoral adviser Prof. Dr. Werner Kießling has given me the opportunity for research in various, interdisciplinary fields of computer science, which was a pleasure for me as well as an incentive for this work. Therefore, I am very grateful for his support, for his helpful comments, and for all the fruitful discussions about my research and this thesis.

I also want to thank Prof. Dr. Elisabeth André from the University of Augsburg (Germany) for her support in the very interesting area of emotional human-computer interaction. Also, I am grateful for helpful discussions about this thesis.

My colleagues are a further reason for the joyful work at my department. Thanks to Stefan Holland, Thorsten Ehm, and Sven Döring for the teamwork within the COSIMA project. Special thanks to Prof. Dr. Bernhard Möller and Anna Schwartz.

For helpful comments and for reading a draft version of this thesis I am grateful to Bernd Hafenrichter, Peter Höfner, and Annette Eberle.

Thanks to my friends, parents, and family for patience and support during the whole period of this work, especially to Stefan, Gregor, and Annette.

Table of Contents

1 Introduction....9

1.1 Database search engines – a continuing misery 9
1.2 Quality of a query result....10
1.3 Deficiencies of personalization within e-procurement....10
1.4 Objectives and positioning of this thesis 13

2 Fundamental Preference Concepts Revisited....17

2.1 Preference modeling - foundations....17
2.1.1 Preferences and its engineering....17
2.1.2 Base preferences....19
2.1.3 Complex preferences....21
2.2 Preference query languages....25
2.2.1 The BMO query model 25
2.2.2 Preference SQL 25
2.2.3 Preference XPath....26
2.3 Preference Repository....26
2.3.1 Storage structure....27
2.3.2 Meta model of situations....27

3 Personalized Presentation of Query Results....29

3.1 Design principles and workflow....30
3.2 An intuitive linguistic model for the quality of query results....36
3.3 Quality information for base preferences 37
3.3.1 SCORE preference....38
3.3.2 BETWEEN preference....41
3.3.3 AROUND preference....42
3.3.4 LOWEST preference....44
3.3.5 HIGHEST preference....45
3.3.6 AT_LEAST preference 47
3.3.7 AT_MOST preference....47
3.3.8 EXPLICIT preference 48
3.3.9 LAYERED_m preference 51
3.3.10 POS/NEG preference 52
3.3.11 POS/POS preference 54
3.3.12 NEG preference....55
3.3.13 POS preference....56
3.3.14 ANTI-CHAIN preference 57
3.4 Quality information for complex preferences....58
3.4.1 Quality valuation of a Pareto preference....58
3.4.2 Quality valuation of a prioritized preference 64
3.4.3 Quality valuation of a numerical preference....68
3.4.4 Quality valuation of grouped preferences 69
3.4.5 Calculation of $\text{QUAL}_{P,s}$....69
3.5 Filter criterion “but-only”....78
3.6 Selection criterion for pointing out a search result....80
3.6.1 General selection criteria....80
3.6.2 Presentation criteria in sales scenarios 85
3.7 Valuating results of other search technologies 88

4 Personalized Offer Composition for E-Procurement ... 91

4.1 E-procurement - state-of-the-art ... 91
4.1.1 Automated e-procurement sales process ... 91
4.1.2 Deficiencies of state-of-the-art technology ... 94
4.2 Preference based components ... 95
4.2.1 Technology for a preference based and smart product composition ... 95
4.2.2 Personalized price offer ... 97
4.2.3 Preference Bargainer ... 99
4.2.4 Data integration by means of e-procurement standards ... 102

5 Automated E-Procurement Sales Agent ... 105

5.1 History ... 105
5.2 The prototype COSIMA^{B2B} ... 107
5.3 The Personalization Manager ... 109
5.4 Evaluation ... 111

6 Related Work ... 117

6.1 Query result presentation ... 117
6.1.1 Parametric search ... 118
6.1.2 Fuzzy logic ... 120
6.1.3 Expert systems / knowledge based systems ... 122
6.1.4 Case based reasoning ... 123
6.1.5 E-catalogs ... 124
6.2 Price fixing ... 125
6.2.1 Personalized price offer ... 125
6.2.2 Preference Bargainer ... 126

7 Summary and Outlook ... 127

7.1 Summary and achievements ... 127
7.2 Outlook and future work ... 128

Literature ... 131

List of Figures ... 139

List of Tables ... 141

Appendix A ... 143

Appendix B ... 147

1 Introduction

The idea of procuring goods via internet is simple. A customer expects to have at least the service he or she has when directly contacting a human sales person. That means the customer wants to be treated individually according to his or her needs. But the misery already begins with the first step, the usage of the search engine.

1.1 Database search engines – a continuing misery

The lack of effectiveness of database search engines is as old as database search engines themselves. If there is no perfect match with respect to the search conditions a best alternative must be delivered. Even Amazon[1], the market leader in the B2C (Business-To-Consumer [Hai02]) domains of books and audio CDs, is not able to present a simple alternative to the desired book "Diary for Robin" by the author "James Patterson", although there is a book by this author with the similar title "Diary for Nicholas" (see Figure 1.1). This phenomenon is known as *empty result effect*.

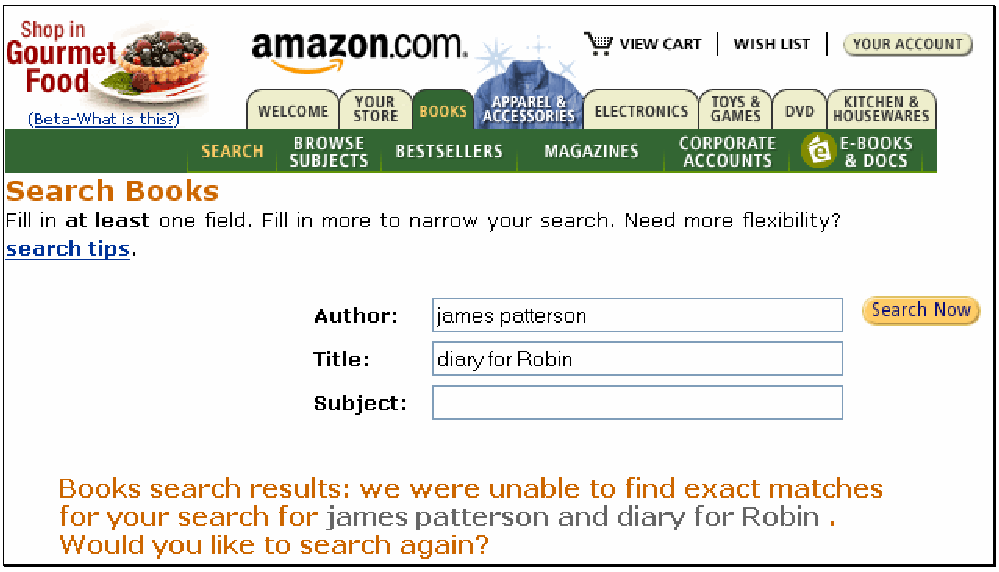


Figure 1.1 Amazon's failing search engine

As a solution, by now most search engines are equipped with the option to combine the search conditions with a logical *or*. E.g. the search engine of the company B2B-Perfect[2] uses this simple technology and promises a powerful search. Especially in B2B (Business-To-Business [Hai02]), where the goods are much more complex than books, the effect is clear, a *flooding effect* with lots of irrelevant results. Fortunately, the misery can be stopped by respecting the customer's search preferences as soft conditions. Modeling preferences as strict partial orders as "A is better than B" semantics ([Kie02]) has been proven to be user intuitive in various internet applications ([KK02]).

---

[1] http://www.amazon.com
[2] http://www.b2b-perfect.com

1.2 Quality of a query result

But more than this is necessary to provide a similarly good service as a human sales person. Since the customer wants to understand why the presented results are the best available ones for him, it is necessary to know about the quality of the query results with respect to the customer's wishes. Normally, to be convinced of the presented preselection, as stated by the sales psychologist Becker ([Bec98]), the customer wants an impression about the search quality of the presented query results as decision support. Some advanced search engines deliver ad hoc and very often not intuitively computed alternatives. The presentation of these results is often equipped with a, for human not comprehensible, valuation that aims to tell how close the result is to the search conditions. E.g. the search engine for the documentation of the Oracle[3] database system scores the results with values between 1% and 100% and orders the results descending. Yet there are no arguments and no explanation to convince the customer regarding the results.

The search engine used by the portal of the scientific association ACM[4] does a little better by valuating the results within five categories (see Figure 1.2), which is the intuitive number of what a human being normally differentiates according to Zadeh, the founder of fuzzy logic theory ([Zad73]). But the user of this search engine can hardly understand why he gets ratings of the second and fourth category when searching only for the word “Kießling”. Taking a closer look into the first resulting paper shows that the work of Kießling is often cited which seems to be the reason for the second highest valuation. But it is not comprehensible why the second paper only gets a rating of the fourth category although Kießling is coauthor of the paper and his work is also cited very often therein. The problem of these valuation approaches lies in the non-personalized and non-situated measurement of key data used in such technologies which obviously fail. A further problem is that all users are treated equally. In this example perhaps one user focuses the author, another user considers citations.

A further and for e-procurement very interesting aspect is that in a sales scenario of course the better the search result, the better is the psychological advantage of the vendor. Therefore, an internet store should have the information about the quality of the search results for being able to provide a good reasoning when offering the results. Each of the well known sales psychology models ([Nic66, HS69, EBK78, Han72]) emphasizes that the knowledge about the quality of the offered goods with respect to the customer's preferences is a major factor for a sales dialog and for consumer choice behavior. The preferences of each customer of course differ. They are even different for one customer in various situations, e.g. someone may in general prefer the color green, but not when considering the color of a car.

1.3 Deficiencies of personalization within e-procurement

E-procurement is the B2B (Business-To-Business) purchase and sale of supplies and services over the internet. The process of searching, presenting and offering goods is a core process of e-

[3] http://www.oracle.com
[4] http://www.acm.org

procurement ([ÖFA01]). Non-personalized state-of-the-art search engines prevent an effective application flow of the interface between vendor and customer. Therefore, many e-procurement ordering portals do even not provide a search engine. The ordering portal only helps customers who exactly know what to buy. E.g. Hilti[5], the world market leader in construction technology, only provides hierarchically listed product catalogs (see label 1 in Figure 1.3), where a customer must manually search through the e-catalog. Thereby, the customer does not know whether there is a desired product or whether the one he possibly found is the best matching alternative result with respect to his search preferences. Only if the customer knows exactly about the existence of a desired product and its item number, he can search for it via one click (see label 2 in Figure 1.3). If he needs any help, he only can get instructions via phone (see label 3 in Figure 1.3). And moreover if the customer finally wants to know about a price for the arranged shopping cart, he only can send the shopping basket via e-mail and wait for an offline calculated personalized offer (see label 4 in Figure 1.3).

Figure 1.2 Quality of search results at ACM

Due to enormous costs of a human sales agent, the demand for effective automation is very high. Both, vendor and customer party would profit from such a technology. The customer would be able

[5] http://www.hilti.com

to efficiently order 24 hours a day and he would immediately receive a personalized price. Moreover, the vendor would be able to grant a higher discount, because he can reduce the costs for human sales agents.

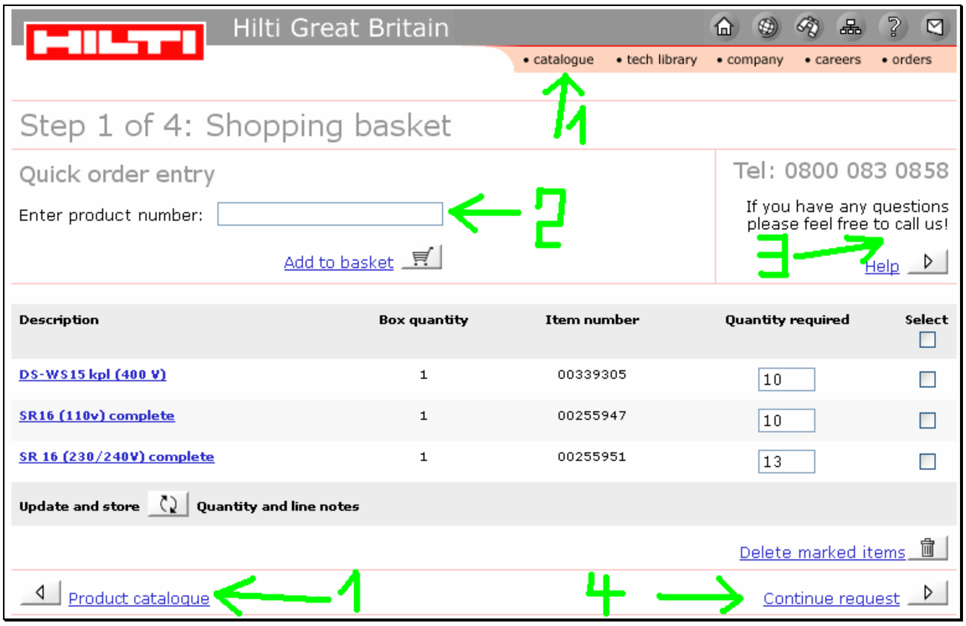


Figure 1.3 Not personalized e-procurement

For small and middle-sized deals with high personnel costs, automation is the logical consequence. According to a general rule in the B2B sector, 80% of the customers cause 20% of turnover and profit, but also cause 80% of the costs. Vice versa with 20% of the customers 80% of the profit is obtained and only 20% of the costs are induced by those customers (Figure 1.4).

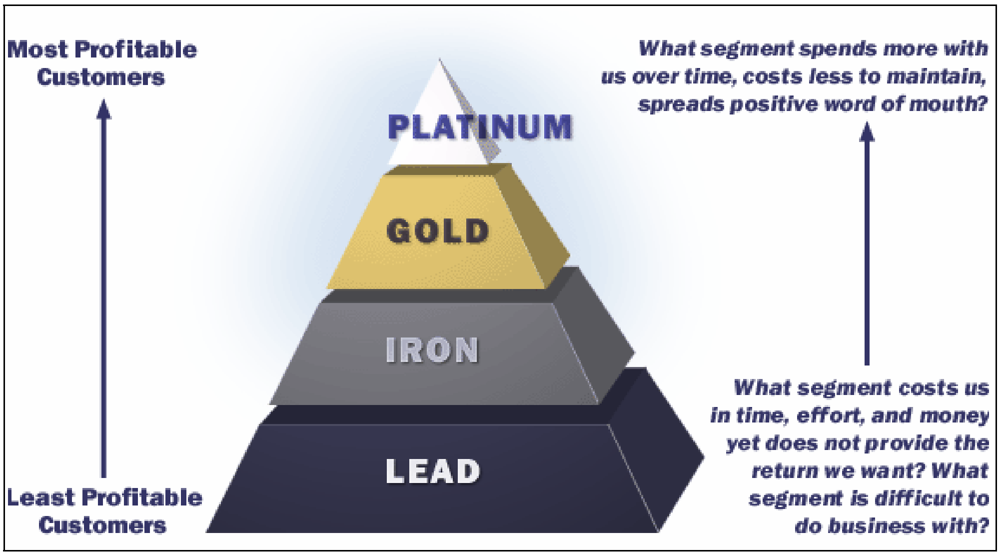


Figure 1.4 B2B customer pyramid - source: ITSMA[6]

[6] http://www.itsma.com

A huge effort of personalization ([RSG01, KBK03]) with a flexible, component based, intuitive, and semantics based underlying model is necessary to make an electronic sales agent no longer a pie in the sky.

1.4 Objectives and positioning of this thesis

The fundamental work about preferences in databases introduced by Kießling ([Kie02, Kie04]) has led the basis for cooperative database search engines. To make search results in general and e-procurement technology in particular more comprehensible and therefore more effective for the customer this interdisciplinary thesis will be based on the semantically rich preference model of [Kie02, Kie04]. In these foundations, preferences are modeled as strict partial orders in "A is better than B" semantics. Novel research aspects and engineering contributions, which will help to handle the above mentioned drawbacks, are the following:

1. **Personalized quality valuation and presentation of a query result**
   The knowledge about the quality of the results with respect to the search preferences is an issue with regard to several aspects. E.g. the quality of a search result may support a customer in his decision which search results might be relevant for him. A further example is that during a sales dialog, the quality of a search result delivers arguments for a vendor. In this thesis a novel personalized and situated quality valuation for query results will be introduced. Based on a human comprehensible linguistic model, an intuitive and universal framework for valuations of query results will be defined. The design of the framework also respects filter conditions stating which results to present and which to hide. A presentation preference applied to the query result set determines which results are predestined to be especially pointed out to the customer. This framework will enable a search engine to proactively present search results by respecting an underlying strategy, e.g. a special sales strategy ([Rac89, HS98]).

2. **Personalized offer composition**
   To effectively automate the e-procurement ordering process for business transactions of small or middle size, a personalized business process will be modeled. In this preference technologies will be used, including two new technologies, designed for price fixing in e-commerce that allow a personalized treatment for each customer in different situations. The first new technology will present a flexible discount framework. The second one will be based on a framework for multi-objective preference based bargaining. For modeling a personalized automatic offer composition current widespread IT product standards like BMEcat and eCl@ss will be exploited.

3. **Engineering of an e-sales agent**
   With COSIMA^{B2B} a deeply personalized and situated prototype of an emotional sales agent will be presented. This autonomous sales agent is based on the above mentioned effective personalized offer composition. An evaluation will give evidence that the modeled frameworks are applicable. Moreover, it will be shown that sales strategies applied by a computer instead of a human also work effectively. Under usage of the above mentioned novel

preference technologies as well as of already existing preference technology components COSIMA^{B2B} shows a real world use case how to effectively reduce costs in the fields of e-procurement.

Several different fields of research had to be regarded for this interdisciplinary work. The following diagram (see Figure 1.5) points out how to position and classify this thesis.

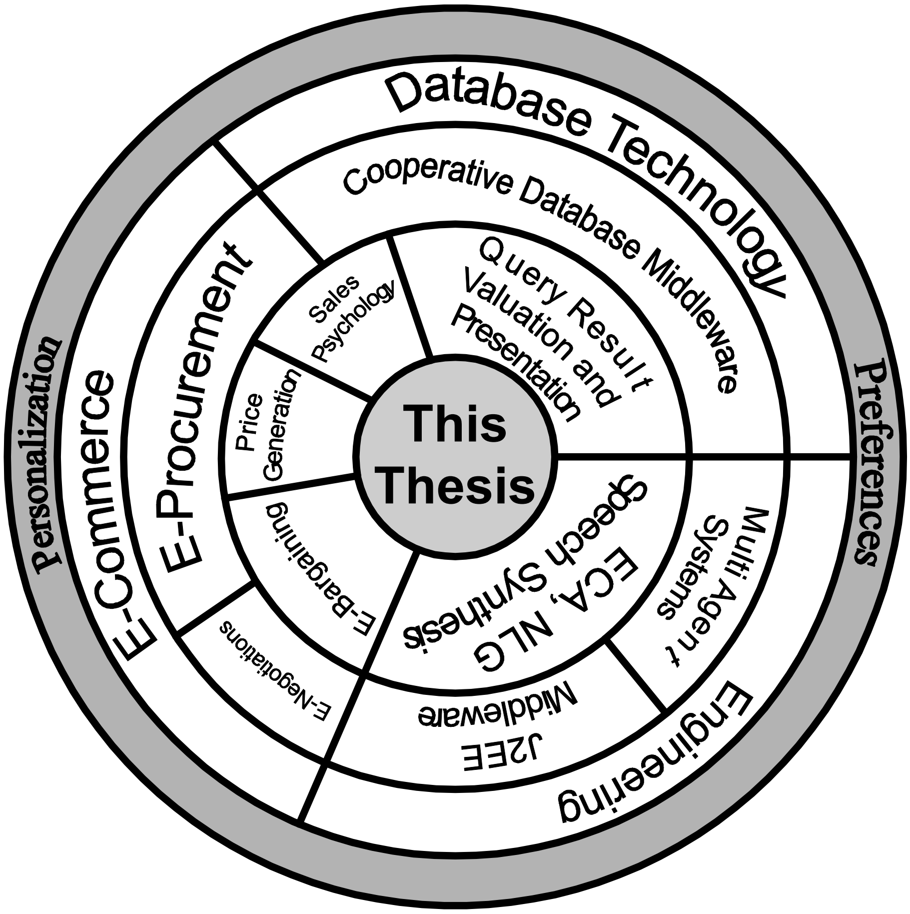


Figure 1.5 Fields of research of this thesis

The valuation and presentation of query results constitute the main part. But also aspects from sales psychology are integrated into cooperative databases (see chapter 3). Sales psychology and results from price fixing as well as improvements in e-bargaining bring novel contributions for two fields of e-commerce, namely e-negotiations and e-procurement (see chapter 4). And, last but not least, there is a large engineering aspect (see chapter 5). A multi-agent platform based on the FIPA[7] ([CP03]) standard is combined with a component based J2EE middleware platform ([HL03]). In that platform additional components for a better human-computer interaction are integrated, namely an embodied character agent ([AR00]), a natural language generation component ([RD97]), and a text-to-speech synthesis technology ([Dut96]). Moreover, interfaces are provided for further technologies like speech and mimic recognition.

[7] http://www.fipa.org

A personalized approach naturally needs a lot of information. Hence, to apply the results of this thesis, a large amount of knowledge is necessary. The search preferences of a customer in a specific situation are needed as well as knowledge about his quality sensation. The detection of this knowledge is not part of this work, but there will be hints how to solve this separate problem. Search preferences will be considered for numerical and categorical values. For full-text search preferences see [LK02]. In terms of presentation the contribution of this thesis concerns the computation of valuable quality information for a good reasoning and the ordering of the results within the presentation. Moreover, criteria telling which results especially to point out will be given to admit a special presentation strategy with the aim to guide the customer. Not of concern are issues about layout or usability of the presentation. Regarding aspects of sales psychology, arguments for a smart presentation will be provided. Moreover, for aspects of sales strategies with respect to the proactive presentation of results, flexibly combinable selection criteria will be developed. However, the focus is not on a complete realization of a sales psychological model. The essential issues of this thesis are:

- How to valuate the quality of a search result so that it is comprehensible for a human and that the human realizes the valuation as appropriate in a specific situation?
- Which results should be presented to a user and which should be hidden?
- In which order should the results be presented; which results are predestined to be pointed out proactively?
- How can major aspects of sales strategies regarding the search result characteristics flexibly be regarded?
- Is a computer system able to replace a human sales agent; is it possible to automate the e-procurement sales process, namely to generate an automatic offer composition?
- Is a computer system able to effectively apply sales strategies to a human customer?

To reach the objectives of this work the proceeding will be as follows. At first the basic search preferences of the user must be considered to compute a comprehensible and correct quality. The preference model of [Kie02, Kie04] provides many basic constructs for formulating most cases in an intuitive and extensible manner. Therefore, these constructs are a suitable foundation for valuating base preferences of a human user. With that knowledge, the combination of base preferences can be valuated and therefore a reasonable overall quality valuation can be performed. The quality of a search result is one major aspect of sales strategies and can be exploited for this purpose.

With that so far and some further preference based technologies an automatic offer composition can be designed. Some aspects of data integration will be elaborated. Finally, performance tests with a first prototype of an automated sales agent will show the applicability. Moreover, some experiments with test customers will show that a computer system is able to effectively apply sales strategies to a human customer.

2 Fundamental Preference Concepts Revisited

To handle user preferences, a comprehensive and theoretical well-founded framework is necessary for a flexible and intuitive usage. Therefore, the foundations of Kießling ([Kie02, Kie04]) for preferences in databases are briefly repeated for this work. They are the basis for an intuitive and flexible valuation of search results as well as for further preference based technologies for e-procurement. For discussion of different preference models and their deficiencies regarding intuition see [Kie02, AW00, GL94, Cho02, Cho03]. Under the motto "It's a Preference World" at the chair for databases and information systems of the University of Augsburg, several further technologies have already been engineered which are helpful for this thesis. They are briefly described in this chapter.

2.1 Preference modeling - foundations

People naturally express their wishes in terms like "I like A better than B". This kind of preference modeling is universally applied and understood by everybody. People are intuitively used to deal with such preferences, in particular with those that are not expressed in terms of numerical scores. But there is also another part of real life primarily concerned with sophisticated economical or technical issues, where numbers do matter. One can easily recognize that numerical ranking can be subsumed under this heading, too. As shown and designed in [Kie02, Kie04], these wishes can be formulated as strict partial orders and can even be engineered to complex, multidimensional preference constructs without loss of intuitive semantics.

2.1.1 Preferences and its engineering

In [Kie02, Kie04] a preference is formulated on a set of attribute names with an associated domain of values. When combining preferences P_1 and P_2, the attributes of P_1 and P_2 may overlap. This allows multiple preferences to coexist on the same attributes. This generality is due to the design principle that conflicts of preferences must be allowed in practice and not be considered as a bug.

Definition 2.1 Domain values of a set of attributes

Let $A = \{A_1, A_2, \ldots, A_k\}$ denote a non-empty set of attribute names A_i associated with domains of values $dom(A_i)$, $1 \leq i \leq k$. The domain of A is defined as $dom(A) := \times_{A_i \in A} dom(A_i)$. *

Definition 2.2 A preference as strict partial order

Given a set A of attribute names, a preference P is a strict partial order $P := (A, <_P)$, where $<_P \subseteq dom(A) \times dom(A)$. The fact that elements x, y are unordered according to P is denoted as $x \parallel_P y$. *

Being a strict partial order means that $<_P$ is irreflexive and transitive and thus also asymmetric. The intension is that

"$x <_P y$" is interpreted as "I like y better than x".

Since preferences reflect important aspects of the real world a good visual representation is essential and can be given by the following 'better-than' graph.

Definition 2.3 'Better-than' graph, quality notions

In finite domains a preference P can be drawn as a directed acyclic graph G, called the 'better-than' graph, which is also known as Hasse diagram ([DP90]). Given G for P the following simple quality notions between values x, y in G are defined:

a) $x <_P y$ if y is predecessor of x in G.
b) Values in G without a predecessor are maximal elements of P (max(P)), being at level 1.
c) x is on level j if the longest path from x to a maximal value has j-1 edges.
d) If there is no directed path between x and y in G then x and y are unranked. *

Complex wishes are abundant in daily, private, and business life, even those concerning several attributes. In [Kie02, Kie04] a powerful and orthogonal framework is given that supports the inductive combination of single preferences into more complex ones. This model is the key towards systematic preference engineering.

The goal is to provide intuitive and convenient ways to inductively construct a preference $P := (A, <_P)$. P is specified by a so-called preference term that fixes the attribute names A and the strict partial order $<_P$. One distinguishes between atomic preference terms, the base preferences, and complex preferences. Since each preference term represents a strict partial order it is identified with a preference P.

Definition 2.4 Preference term

Given preference terms $P_1, P_2, \ldots P_w$, a preference term is one of the following:

- Any base preference: $P := \text{basepref}_i$
- Any complex preference P gained by applying one of the following preference constructors:
  - Pareto preference: $P := P_1 \otimes P_2 \otimes \ldots \otimes P_w$
  - Prioritized preference: $P := P_1 \,\&\, P_2 \,\&\, \ldots \,\&\, P_w$
  - Numerical preference: $P := \text{rank}_F(P_1, P_2, \ldots, P_w)$ *

In the sequel the already provided base preferences of [Kie02, Kie04] are enhanced by two new base preferences "AT_LEAST" and "AT_MOST" and are described as well as the three mentioned complex preferences. For a comfortable and intuitive preference engineering the given preference constructors, which in fact are preference templates, are described, starting with the non-numerical base preferences, then the numerical base preferences, and concluding with the mentioned complex preferences.

2.1.2 Base preferences

People often have sympathy or antipathy for values of a categorical attribute, e.g. someone likes green clear stock boxes but wants to avoid black ones. The following preference constructs enable one to easily model the most frequently occurring expressions (POS, NEG, POS/NEG, POS/POS, LAYERED_m, ANTI-CHAIN) as well as a totally flexible and powerful expression possibility with the EXPLICIT preference.

Definition 2.5 Categorical or non numerical base preferences

- Given a finite POS-set $\subseteq$ dom(A) of favorite values, P := POS(A, POS-set) is a **POS preference** if for all x, y $\in$ dom(A): $x <_P y \Leftrightarrow x \notin \text{POS-set} \wedge y \in \text{POS-set}$.
- Given a finite NEG-set $\subseteq$ dom(A) of disliked values, P := NEG(A, NEG-set) is a **NEG preference** if for all x, y $\in$ dom(A): $x <_P y \Leftrightarrow y \notin \text{NEG-set} \wedge x \in \text{NEG-set}$.
- Given a finite POS-set $\subseteq$ dom(A) of favorite values and a disjoint finite NEG-set $\subseteq$ dom(A) of disliked values, P := POS/NEG(A, POS-set; NEG-set) is a **POS/NEG preference** if for all x, y $\in$ dom(A):
  $x <_P y \Leftrightarrow (x \in \text{NEG-set} \wedge y \notin \text{NEG-set}) \vee (x \notin \text{NEG-set} \wedge x \notin \text{POS-set} \wedge y \in \text{POS-set})$.
- Given a finite POS_1-set $\subseteq$ dom(A) of favorite values and a disjoint finite POS_2-set $\subseteq$ dom(A) of alternative values, P := POS/POS(A, POS_1-set; POS_2-set) is a **POS/POS preference** if for all x, y $\in$ dom(A):
  $x <_P y \Leftrightarrow (y \in \text{POS}_1\text{-set} \wedge x \notin \text{POS}_1\text{-set}) \vee (y \in \text{POS}_2\text{-set} \wedge x \notin \text{POS}_1\text{-set} \wedge x \notin \text{POS}_2\text{-set})$.
- Let $L = (L_1, \ldots, L_{m+1})$, $m \geq 0$, $m \in \mathbb{N}$, be an ordered list where L is a partition of dom(A) and exactly m out of these m+1 sets are given as finite enumerations of values from dom(A). The remaining set is specified as 'other values'. The function layer: dom(a) $\rightarrow \mathbb{N}$ is defined as follows: for i $\in$ {1, …, m+1}, for all v $\in L_i$: layer(v) := i. P := LAYERED_m(A, L) is a **LAYERED_m preference** if for all x, y $\in$ dom(A):
  $x <_P y \Leftrightarrow \text{layer}(y) < \text{layer}(x)$.
- Given an E-graph = {(val_1, val_2), …} representing a finite acyclic 'better-than' graph as described in Definition 2.3 and a set G of all $\text{val}_i \in$ dom(A) occurring in E-graph. A strict partial order E = (G, $<_E$) is induced as follows:
  - (val_i, val_j) $\in$ E-graph implies $\text{val}_i <_E \text{val}_j$
  - $\text{val}_i <_E \text{val}_j \wedge \text{val}_j <_E \text{val}_k$ imply $\text{val}_i <_E \text{val}_k$

  P := EXP(A, E-graph) is an **EXPLICIT preference** if
  $x <_P y \Leftrightarrow x <_E y \vee (x \notin \text{range}(<_E) \wedge y \in \text{range}(<_E))$,
  where $\text{range}(<_E) := \{y \in \text{dom}(A) \mid \exists z \in \text{dom}(A): (y, z) \in \text{E-graph} \vee (z, y) \in \text{E-graph}\}$.
- A preference P is an **ANTI-CHAIN preference** if $x <_P = \emptyset$. The anti-chain on an attribute A is denoted as $A^{\leftrightarrow}$. *

Example 2.1 POS/NEG preference

When Marge plans to go on a journey to visit a city she **prefers** Minnesota but she **avoids** Los Angeles and Orlando. These simple preferences of likes and dislikes can easily be formulated in the POS/NEG preference POS/NEG := (cities, {Minnesota}; {Los Angeles, Orlando}). This means, she likes Minnesota better than all other cities, but within these cities Los Angeles and Orlando are her worst alternatives. ⌂

Example 2.2 EXPLICIT preference

Bart's preferences regarding to cities are more explicit. He **likes** Denver **more than** San Antonio. He **prefers** Sacramento **over** Memphis and Houston. Even San Antonio is **better than** Houston. But Houston is **better than** Cleveland. Memphis is **better than** New Jersey. This preference can be expressed as EXP(cities, E-graph), where E-graph is the graph illustrated in Figure 2.1.

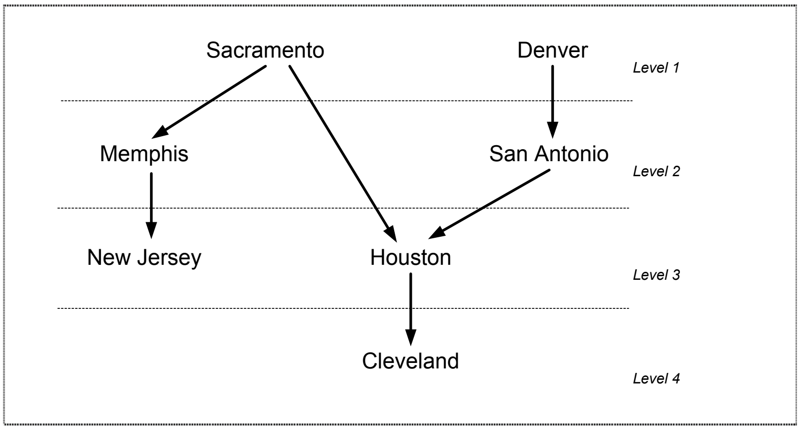


Figure 2.1 E-graph for Bart's EXPLICIT preference

⌂

Considering domains like measures, price, or age that support a total comparison operator '<' and a subtraction operator '–', the following numerical base preferences (LOWEST, HIGHEST, AROUND, BETWEEN, AT_LEAST, AT_MOST) are the suitable constructs. For a flexible arithmetic preference engineering the SCORE preference allows formulating any kind of numerical ranking function. Instead of the discrete 'level' function of Definition 2.3, continuous 'distance' functions are employed, defined on '<' and '–' for numerical base preferences.

Definition 2.6 Numerical base preferences

- P := LOWEST(A) is called **LOWEST preference** if for all x, y ∈ dom(A):
  $x <_P y \Leftrightarrow x > y$.
- P := HIGHEST(A) is called **HIGHEST preference** if for all x, y ∈ dom(A):
  $x <_P y \Leftrightarrow x < y$.
- Given a desired value z ∈ dom(A), for all v ∈ dom(A) the distance is defined as
  distance(v, z) := abs(v – z).
  P := AROUND(A, z) is called **AROUND preference** if for all x, y ∈ dom(A):
  $x <_P y \Leftrightarrow$ distance(x, z) > distance(y, z).

- Given an interval $[low, up] \in dom(A) \times dom(A)$ of desired values, for all $v \in dom(A)$ the distance is defined as distance(v, [low, up]) := if $v \in [low, up]$ then 0
  else if $v < low$ then $low - v$ else $v - up$.
  P := BETWEEN(A, [low, up]) is a **BETWEEN preference** if for all $x, y \in dom(A)$:
  $x <_P y \Leftrightarrow$ distance(x, [low, up]) > distance(y, [low, up]).
- Given an at least desired value $z \in dom(A)$, for all $v \in dom(A)$ the distance is defined as
  distance(v, z) := if $v \geq z$ then 0 else $z - v$.
  P := AT_LEAST(A, z) is called **AT_LEAST preference** if for all $x, y \in dom(A)$:
  $x <_P y \Leftrightarrow$ distance(x, z) > distance(y, z).
- Given an at most desired value $z \in dom(A)$, for all $v \in dom(A)$ the distance is defined as
  distance(v, z) := if $v \leq z$ then 0 else $v - z$.
  P := AT_MOST(A, z) is called **AT_MOST preference** if for all $x, y \in dom(A)$:
  $x <_P y \Leftrightarrow$ distance(x, z) > distance(y, z).
- Given a scoring function f: $dom(A) \rightarrow \mathbb{R}$ and the familiar 'less-than' order '<' on $\mathbb{R}$.
  P := SCORE(A, f) is called **SCORE preference** if for all $x, y \in dom(A)$:
  $x <_P y \Leftrightarrow f(x) < f(y)$. *

Example 2.3 BETWEEN preference

Ned also likes trips to cities. But his preference depends on the number of people living in these cities. He **prefers** cities with a population **between** 600000 and 850000, because Ned believes a city with less than 600000 people does not have enough cultural facilities and in a city with more than 850000 people there is too much criminal energy. Formally expressed the preference reads BETWEEN(population, [600000, 850000]). ⌂

The just mentioned base preference constructors can be arranged into the taxonomic hierarchy represented in Figure 2.2. The AROUND, HIGHEST, and LOWEST sub-constructors can be obtained by identifying low = up =: z and by choosing z as the finite supremum and infimum of dom(A), respectively. The AT_MOST and AT_LEAST sub-constructors can be obtained by identifying low as the finite infimum and up as the finite supremum of dom(A), respectively.

2.1.3 Complex preferences

Preferences are often more complex than described above. Therefore, in [Kie02, Kie04] constructors for combining the described base preferences are defined. The accumulation of preferences can be done in the following three manners, namely as Pareto preference, prioritized preference, and numerical complex preference. The Pareto-optimality principle has been studied intensively for multi-attribute decision problems in the social and economic sciences ([SB94]). Here it is defined for w = 2 preferences. The generalization to w > 2 is obvious.

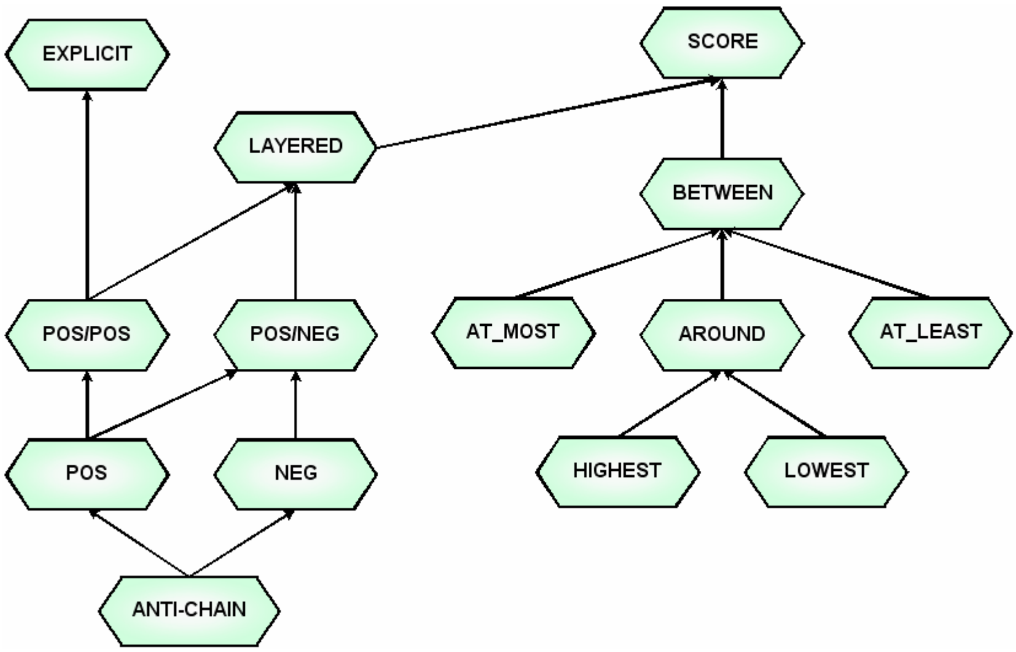


Figure 2.2 Hierarchy of base preference constructors

Definition 2.7 Pareto preference

Given $P_1 = (A_1, <_{P_1})$ and $P_2 = (A_2, <_{P_2})$, for all $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathrm{dom}(A_1) \times \mathrm{dom}(A_2)$ a preference $P := P_1 \otimes P_2$ is called a **Pareto preference** if

$$x <_{P_1 \otimes P_2} y \Leftrightarrow (x_1 <_{P_1} y_1 \wedge (x_2 <_{P_2} y_2 \vee x_2 = y_2)) \vee (x_2 <_{P_2} y_2 \wedge (x_1 <_{P_1} y_1 \vee x_1 = y_1)). \quad *$$

Example 2.4 Pareto preference

Lisa's preferences regarding a journey to a city are a little more complex. She **prefers** cities with a population of **about** 500000 people **and equally important** the city **should be** a state capital, **and equally important** the maximum temperature during winter days **should be at most** 16 degrees Celsius. These three preferences can be expressed as

$P_1 \otimes P_2 \otimes P_3$, where $P_1 :=$ AROUND(population, 500000)
$P_2 :=$ POS(state_capital, {yes})
$P_3 :=$ AT_MOST(degrees Celsius, 16).

The 'better-than' graph for four sample cities with the relevant data is given in Figure 2.3.

If a preference is more important than another then the following prioritized preference supports the intuitive modeling. It is defined for w = 2 preferences. The generalization to w > 2 is obvious.

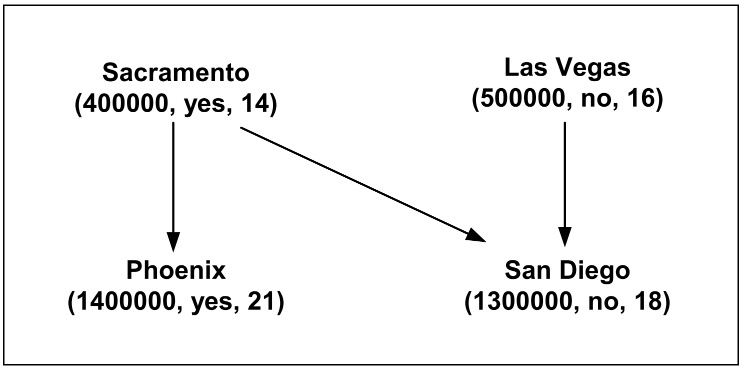


Figure 2.3 'Better-than' graph for Lisa's Pareto preference

⌂

Definition 2.8 Prioritized preference

Given $P_1 = (A_1, <_{P_1})$ and $P_2 = (A_2, <_{P_2})$, for all $x = (x_1, x_2)$, $y = (y_1, y_2) \in dom(A_1) \times dom(A_2)$ a preference $P := P_1 \ \& \ P_2$ is called a **prioritized preference** if

$$x <_{P_1\&P_2} y \Leftrightarrow x_1 <_{P_1} y_1 \vee (x_1 = y_1 \wedge x_2 <_{P_2} y_2).$$ *

Example 2.5 Prioritized preference

Homer **prefers** cities with a Sizzler steak house. This is **more important** than a population **around** 500000 people. This can be expressed as

$P_1 \ \& \ P_2$, where $P_1 := POS(Sizzler, \{yes\})$
$P_2 := AROUND(population, 500000)$.

The 'better-than' graph for three sample cities is illustrated in Figure 2.4.

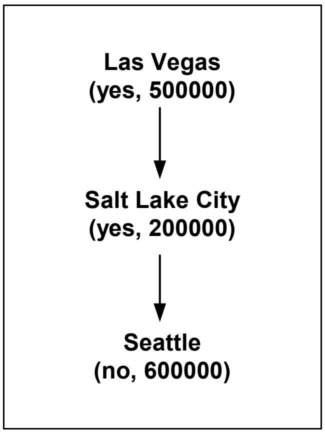


Figure 2.4 'Better-than' graph for Homer's prioritized preference

⌂

If there is the need to accumulate preferences with special weights for each preference involved the following numerical preference provides this model. Numerical preferences build on SCORE preferences. Note that, according to the sub-constructor hierarchy of preferences of Figure 2.2 several other preferences can be formulated as SCORE preferences. The individual scores are accumulated into an overall score by applying a multi-attribute combining function F. Here it is defined for $w = 2$, where the generalization to $w > 2$ is obvious.

Definition 2.9 Numerical preference

Given P_1 = SCORE(A_1, f_1), P_2 = SCORE(A_2, f_2) and a combining function F: $\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. For all $x = (x_1, x_2)$, $y = (y_1, y_2) \in dom(A_1) \times dom(A_2)$ a preference $P := rank_F(P_1, P_2)$ is called a **numerical preference** if $x <_{rank_F(P_1, P_2)} y \Leftrightarrow F(f_1(x_1), f_2(x_2)) < F(f_1(y_1), f_2(y_2))$. *

Example 2.6 Numerical preference

Barnie's preferences about cities are a little bit hard to understand, but he has built up his special rating for cities. He has a SCORE preference P_1 for the population x_1 of a city, where the score function is defined as:

$$f_1(x_1) := \begin{cases} 1, & 1000000 < x_1 \\ 2, & 500000 \le x_1 \le 1000000 \\ 3, & x_1 < 500000 \end{cases}$$

His second preference is about the highest temperature x_2 during winter days. This HIGHEST preference can easily be interpreted as a SCORE preference P_2 with the scoring function $f_2(x_2) = x_2$. For both preferences Barnie formulates the combining function F with $F(f_1(x_1), f_2(x_2)) = 2\ f_1(x_1) + 0.5\ f_2(x_2)$. With the data from above mentioned examples the values for the three sample cities Sacramento, Phoenix, and San Diego are computed as follows:

Sacramento: $F(f_1(400000), f_2(14)) = 2 * 3 + 0.5 * 14 = 13$
Phoenix: $F(f_1(1400000), f_2(21)) = 2 * 1 + 0.5 * 21 = 12{,}5$
San Diego: $F(f_1(1300000), f_2(18)) = 2 * 1 + 0.5 * 18 = 11$

The according 'better-than' graph is given in the following illustration.

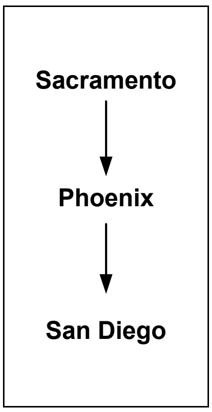


Figure 2.5 'Better-than' graph for Barnie's numerical preference

⌂

2.2 Preference query languages

Preferences are defined in terms of values from dom(A), representing the realm of wishes. In database applications it is assumed that the real world is mapped into appropriate instances that are called database sets. A database set R may e.g. be a view or a base relation in SQL or a DTD-instance in XML.

2.2.1 The BMO query model

Whether preferences can be satisfied depends on the current database contents, capturing the status of the real world. Thus a match-making between wishes and reality has to be accomplished. To this purpose the BMO ("Best Matches Only") query model has been introduced in [Kie02, Kie04]. Assume a schema $R(A_1 : dom(A_1), \ldots, A_m : dom(A_m))$ and consider a preference $P(A, <_P)$, where $A \subseteq \{A_1, \ldots A_m\}$. Let R[A] denote the projection $\pi_A(R)$. Then a preference selection is defined as follows.

Definition 2.10 BMO result

The result of the soft selection on the relation R with a user preference P is specified as

$$BMO := \sigma[P](R) := \{t \in R \mid t[A] \in \max(P^R)\},$$

where $P^R := (R[A], <_P)$ is a database preference. *

Since it only contains the maximal values, the BMO result set is a preselection of unordered result tuples. That means for all tuples $t, t' \in$ BMO: $t[A] \parallel_P t'[A]$.

Definition 2.11 Grouped preferences

Given $P := (B, <_P)$, an attribute set A, and a database preference P^R, then a preference query with grouping is defined as

$$\sigma[P \text{ groupby } A](R) := \sigma[A^{\leftrightarrow}\&P](R),$$

where $A^{\leftrightarrow} := (A, \emptyset)$ denotes the special case of a so called anti-chain preference, which means that all pairs of values of A are unordered. *

Within the "Preference World" two of the most well-known standard database query languages are extended by soft conditions based on the preference model described above, one for relational databases and one for XML structured data.

2.2.2 Preference SQL

The search engine Preference SQL ([KK02]) is an extension of standard SQL that provides, additionally to hard constraints, denoted as WHERE-clause in SQL, the usage of soft conditions by using the keyword PREFERRING. In this, firstly, all hard filter conditions of the WHERE-clause are considered. Secondly, the PREFERRING clauses are evaluated according to the above mentioned preference model. Only the best matches (BMO) as defined in Definition 2.10 are delivered.

Preferences as partial orders have been shown to be compatible with relational and deductive database technologies ([KKTG95]). Possibilities of algebraic query optimization of preference conditions have been studied in [Haf03, KH02]. Preference SQL supports various base preferences as well as the Pareto preference ('AND'-operator) and the prioritized preference (','-operator).

2.2.3 Preference XPath

In e-commerce XML ([BPSM00]) has become a very important standard for storing, presenting, and exchanging data. XPath ([CD99]) has become one of the important languages for requesting data in XML documents. Like all other query languages XPath only supports hard conditions and is therefore not able to handle user wishes carefully. XPath has been extended with strict partial order preferences ([KHFH01]) according to the above mentioned preference model. The resulting XML query language Preference XPath is fully XPath compatible but additionally supports soft condition queries. Instead of the XPath syntax "[*hard conditions*]", soft conditions are captured by "#[*soft conditions*]#".

Example 2.7 Soft conditions with Preference XPath

A travel agency works with an XML-database, where the data is structured as follows:

```
<CITY>
  <NAME>Las Vegas</NAME>
  <POPULATION>500000</POPULATION>
  <STATE_CAPITAL>no</STATE_CAPITAL>
  <TEMPERATURE>16</TEMPERATURE>
  <SIZZLER>yes</SIZZLER>
</CITY>
<CITY>
 …
</CITY>
```

Table 2.1 XML database excerpt of available cities

Considering Homer's preference of Example 2.5 the corresponding query statement formulated in Preference XPath reads as follows.

```
/CITY #[SIZZLER IN ('yes') PRIOR TO
        POPULATION AROUND(500000)]#
```

Figure 2.6 Sample Preference XPath statement

⌂

2.3 Preference Repository

To manage user preferences with the Preference Repository ([Hol03]) an appropriate XML based storage structure has been defined by Holland including situational aspects.

2.3.1 Storage structure

Additional to the preference model of [Kie02, Kie04] in this structure the origin of a preference can be specified. This can e.g. be the explicit information given by the user about his preferences, gained from preference queries using Preference SQL or Preference XPath, or from Preference Mining techniques ([HEK03, Hol03]). In the following Figure 2.7 Lisa's Pareto preference of Example 2.4 is stored. Note that also the situation with according condition is specified when this preference is valid. In this example it depends on the season whether Lisa has this preference about the temperature.

```
<PreferenceData name="LisaCityTripInWinter">
   <TimeStamp dateTime="2003-03-04T10:51:27"/>
   <Source prefSource="ExplicitUserInformation"/>
   <Situation>
      <Condition key="Season" value="winter"/>
   </Situation>
   <Preference>
      <Pareto>
         <Preference>
            <AROUND att="POPULATION" val="500000"/>
         </Preference>
         <Preference>
            <POS att="STATE_CAPITEL">
             <POSSet>
               <Value val="yes"/>
             </POSSet>
            </POS>
         </Preference>
         <Preference>
            <AT_MOST att="DEGREES CELSIUS" val="16"/>
         </Preference>
      </Pareto>
   </Preference>
</PreferenceData>
```

Figure 2.7 Sample Preference Repository

2.3.2 Meta model of situations

To handle preference oriented situations, in [Hol03, HK04] a meta model has been designed based on the well-known entity-relationship notation (see Figure 2.8). As defined in the Oxford Advanced Learner's Dictionary ([Hor00]) a situation consists of "all the circumstances and things that are happening at a particular time and in a particular place". Therefore, in the model of [Hol03] these three subjects have an impact on the situation:

1. Personal situation-specific information
   Personal information, like the physical or psychological condition, emotions, or the current role, has a big influence on the situation of a human.

2. Spatial-temporal framework
   As described in [RZ97], the spatial-temporal frame is an important aspect for situation models, since a user's preferences may change over time or may depend on the current location. For example, as described before, the preference according to a color may be different at a car shop in contrast to when buying furniture.
3. Surrounding influences
   Typically, a user does not act in a closed world but is interacting with his environment. Therefore it is important to know which surrounding influences are relevant for the user's situation. Such influences can be people, events, weather condition, etc.

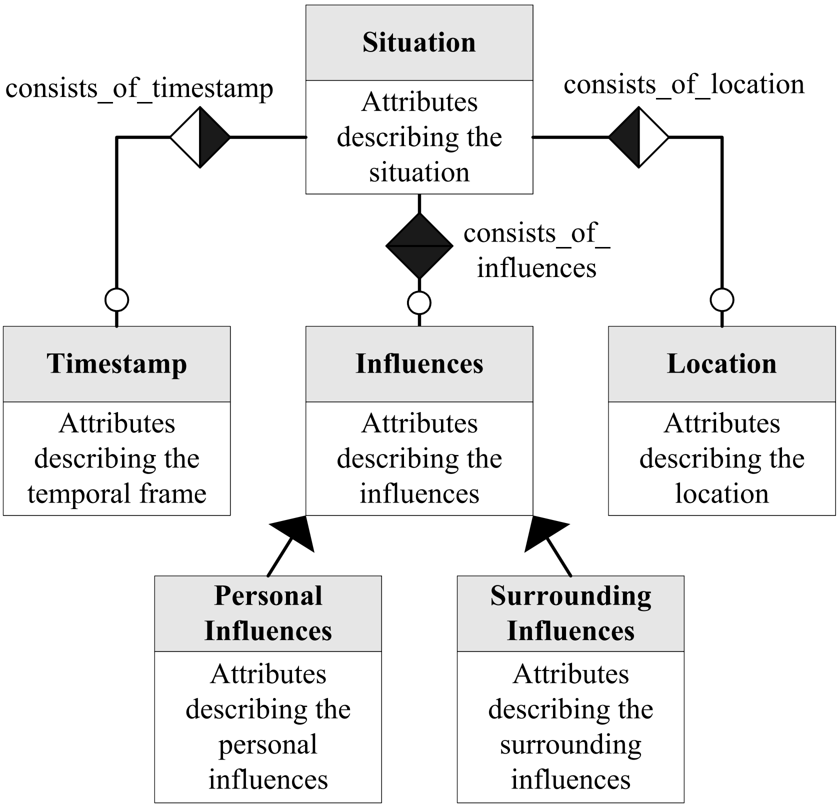


Figure 2.8 Meta model for situations

3 Personalized Presentation of Query Results

Operating the search engine of a database is a very powerful position and decisive question of design ([DFN01]). E.g., considering the power of the decision which results to present first at Google's[8] search engine, the first listed links have much more visits, with all the positive consequences of this. When setting up the search engine in a database, there are five steps ([Har02]):

1. Preparing the sources 2. Indexing 3. Interpreting the query 4. Searching	XML or SQL database delivers BMO results
5. Presentation of the results	BMO as temporary relation in an XML database

Steps 1 and 2 are parts of the database management system. Step 3 could be qualitatively solved by preference search technologies like Preference SQL or Preference XPath (see section 2.2), to avoid the *empty result effect* and the *flooding effect with irrelevant results*. Only best matches are presented. The search within the database – step 4 – is done by the database system.

The importance of step 5 is still nearly ignored. Especially in a sales scenario, it is very important which results to present at all, which alternatives to present, which ones to put in front, or even to recommend. Here also the question is whether the preferences of the presenter are also respected. There are some ad hoc attempts to cover these problems, e.g. the providers of e-catalog technology Poet[9] and EMS Media[10] allow specifying so called favorites by flags in the database. Whenever favorites are in the result set they are placed at the top of the result table. But that, of course, is not a customer-tailored approach and respects the preferences of the presenter only statically. There is no respect to any customer preferences or to aspects of specific situations.

One major and still unsolved key issue in these difficulties is about the quality of the search result with respect to the customer preferences. The knowledge about the qualities of search results is absolutely necessary to do a good reasoning when presenting results.

In this chapter a novel personalized and situated presentation framework is introduced. A valuation of query results is, of course, necessary only for advanced search engines that are able to present alternatives when there are no perfect matches. In the sequel, the BMO result model as described in Definition 2.10 is applied, because it is very user intuitive and has been shown to be suitable for a cooperative product search in various applications ([KK02]). At the end of this chapter, it is explained that this presentation framework is compatible with other advanced search technologies, too.

[8] http://www.google.com
[9] http://www.poet.com
[10] http://www.ems-ag.de

3.1 Design principles and workflow

For a successful deal, the product presentation is a decisive factor. An example of an advanced search result presentation is shown in the following small scenario.

Example 3.1 Sample scenario of an advanced search result presentation

Marge is a computer dealer and tells her vendor Apu:

> *"I am interested in notebooks. The clock frequency* ***must be at least*** *2 GHz. The order quantity* ***should be around*** *40. It is* ***equally important*** *that the main memory capacity* ***should be at least*** *512 MB-RAM, and it is* ***equally important*** *that the price* ***should be at most*** *1200 €.* ***Equally important****, my* ***preferred*** *manufacturers are Toshiba and Hewlett Packard."*

An answer from the vendor side might be the BMO result set shown in the following Table 3.1.

	Make	CPU-GHz	MB-RAM	Quantity	Price per unit (€)
t_1	Elitegroup	2.0	256	40	1150.--
t_2	Gerion	2.0	374	50	1199.--
t_3	HP	2.2	512	50	1249.--
t_4	Toshiba	2.4	768	40	1378.--

Table 3.1 Sample BMO result set

Even in this not too complex example the customer would be glad about advice and assistance. Additionally, a proactive recommendation of a product supports an effective selling process ([Bän85, Fis87]). A cooperative result presentation could now pick out one result. This could be the Toshiba notebooks, because the vendor's utility function of maximizing the turnover is supported and, besides the hard condition of the CPU-GHz, three of four preferences are perfectly fulfilled. This delivers the following sales arguments:

> *"There are four best matches with respect to your preferences. I recommend the Toshiba notebook position which* ***very well*** *meets your preferences. It* ***perfectly*** *matches your desired manufacturer* ***and*** *order quantity and has even a faster CPU and* ***more*** *MB-RAM as you demanded. I think the moderately higher price is* ***acceptable*** *for this high-quality product."*

Following a different presentation strategy the presenter can pick out the third result and argue:

> *"A good choice would be provided by the third position, the Hewlett Packard with a 2.2 GHz CPU. It* ***perfectly*** *meets your preferences with respect to manufacturer* ***and*** *MB-RAM. I think to order 50 notebooks is* ***acceptable and*** *1249.-- € per notebook is a* ***good*** *price."*

⌂

The first aspect of the following design principles serves to avoid incomprehensible and non-personalized ad hoc attempts of presenting search results with incorrect valuations and disrespect of the preferences of customer and presenter, i.e. the vendor side. Moreover, the second aspect serves to enable advanced search engines to do a cooperative and effective presentation of the search results following the strategy of the presenter:

1. **Consideration of customer preferences as well as vendor preferences**
   From the search to the final presentation the preferences of the customer as well as the preferences of the vendor must be flexibly combinable with respect to the specific situation.
2. **Providing plausible arguments about the quality of the search results**
   For an effective presentation of search results the vendor needs arguments to underpin his decision why to present or point out the specific results. Therefore, it is necessary to know about the search quality with respect to single search conditions as well as about an overall impression of the quality of a search result for the specific situation.
3. **Consideration of filter criteria applied to the search result set**
   In various cases not all computed alternatives should be shown to the customer. The reason may be e.g. search quality requirements of the customer or results the presenter only wants to show under specific circumstances and different situations. Therefore it must be possible to apply flexible filter conditions to the search result set.
4. **Semantically well-founded marked favorites within the result set**
   To enable the vendor to point out one or more of the results, these have to be marked following flexible and situated criteria such as e.g. a utility function of a vendor or the result quality with respect to a customer's search preferences.
5. **Universal interface for the usage of the presentation information**
   Different platforms or devices should be able to handle the semantic information about the quality and the favorites of a search result in a uniform way. Therefore a universal interchange format must be provided.

Definition 3.1 describes the major steps of the workflow of a personalized advanced query result presentation based on the BMO query model as described above.

Definition 3.1 Workflow of a result presentation

The presentation of BMO query results is defined in seven steps:

1. Query composition with *search preferences*
2. *Preference search*
3. Computing the *quality valuations* of the BMO results
4. Computing *aggregated quality valuations*
5. Applying *"But-only" filter* conditions to the BMO results
6. Marking out favorites via *presentation preferences*
7. Result *presentation* and consideration of customer *feedback*

*

The workflow of Definition 3.1is illustrated in Figure 3.1.

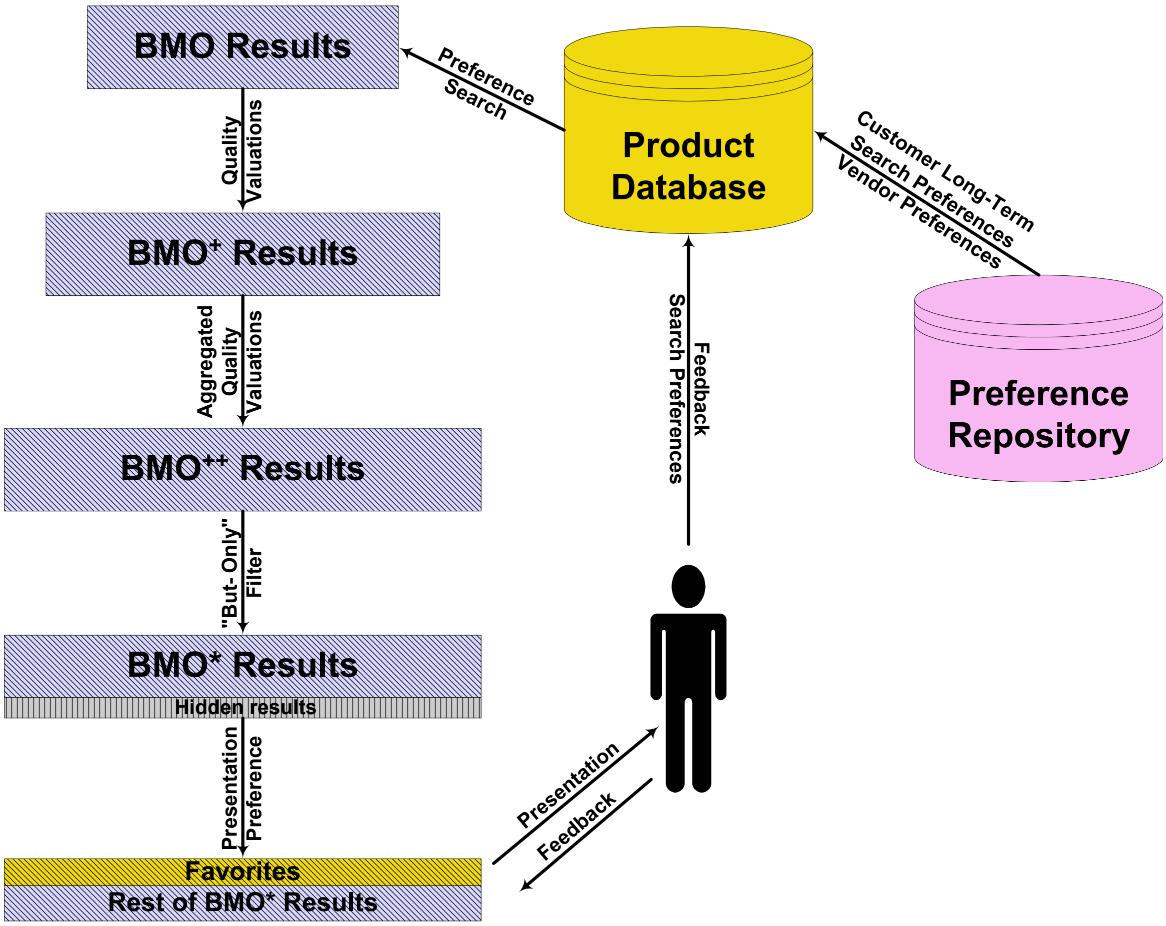


Figure 3.1 Workflow of a BMO result presentation

In the sequel the steps of Definition 3.1 are outlined.

1. Query composition with *search preferences*:
Before starting a qualitative product search the search preferences must be known. The customer's search preferences are e.g. expressed within a search mask composed with his long-term search preferences managed in a Preference Repository (see section 2.3). Also vendor preferences can be regarded which can act as a-priori filters. An intuitive composition of these different preferences can easily be obtained, using the preference model of [Kie02, Kie04] as described in Definition 2.2 and Definition 2.4.

2. *Preference search*:
With a given search preference best matches can be queried from a database with a preference query language (see section 2.2). The results are only best matches according to the BMO model of Definition 2.10. This provides a clearly arranged preselection without bothering the customer by lots of search iterations of refining the search criteria.

3./4. Computing the *quality valuations* of the BMO results and *aggregated quality valuations*:
The knowledge of the quality of a search result with respect to each occurring base preference provides valuable presentation arguments. Because a human normally does not recognize too many

different conditions, each result tuple $t \in BMO$ is mapped via partitioning to a linguistic quality term $v \in V = \{v_1, ..., v_z\}$ for each occurring base preference. With that, a basis is given for an objective quality computation for search results with respect to complex preferences and therefore to an overall quality computation. As discussed before preferences depend on situations. Thus, the following Definition 3.2 specifies the quality function of search results for the underlying situation.

Definition 3.2 Quality function $QUAL_{P,s}$

Let $V := (v_1, ..., v_z)$ be a descending ordered list of linguistic quality terms and let $C(s) := \{C_1(s), \ldots, C_z(s)\}$ be a partition of dom(A) into z parts depending on the situation s. This means there are as many parts $C_j(s)$ in C as linguistic terms v_j in V. Then the quality function $QUAL_{P,s} : BMO \rightarrow V$ for a search result tuple $t \in BMO$ with $t[A] \in dom(A)$ regarding a preference $P := (A, <_P)$ and a situation s is defined as follows:

$$QUAL_{P,s}(t) := \begin{cases} v_1 , & t[A] \in C_1(s) \\ \vdots & \vdots \\ v_j , & t[A] \in C_j(s) \\ \vdots & \vdots \\ v_z , & t[A] \in C_z(s) \end{cases}$$

*

Obviously, $QUAL_{P,s}(x)$ is also well-defined for all elements $x \in dom(A)$. Please note that for a presentation to a customer the possibly involved vendor preferences do not have to be valuated, because this would not provide a presentation argument with respect to the customer.

Of course, when placing several items with different properties into quality categories it is possible that a pair of ordered items is placed into the same category. Yet, it is intuitively not comprehensible that a less preferred element of an ordered pair belongs to a higher quality category than the more preferred element. Even in a BMO result set, where no ordered pairs are included, this aspect is very important. Considering the database excerpt of Table 3.2 and the customer preference $P := HIGHEST(CPU\text{-}GHz) \otimes HIGHEST(MB\text{-}RAM)$.

	Make	CPU-GHz	MB-RAM	Price per unit (€)
t_1	Gerion	2.0	256	1150.--
t_2	Gerion	2.0	374	1199.--
t_3	Gerion	2.2	512	1249.--

Table 3.2 Sample database excerpt

Via online request tuple t_3 is offered to the customer with a quality of v_i. After thinking about the offer he decides to order the notebook but in this moment the notebook t_3 is sold out. The vendor offers t_2 as a further alternative and argues with quality v_j for this result, where $v_i < v_j$. The customer would not understand why the obviously better result regarding his preferences would be valuated worse than the worse alternative. For him the situation has not changed. Therefore, $QUAL_{P,s}$ must fulfill the following postulate of Definition 3.3.

Definition 3.3 Quality postulate

For a given preference P := (A, $<_P$) in a situation s the $QUAL_{P,s}$-function of Definition 3.2 must satisfy that

for all elements t, t' with t[A], t'[A] $\in$ dom(A): t $<_P$ t' $\Rightarrow$ $QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$. *

Depending on the situation unordered result tuples can have different quality valuations. E.g. considering Table 3.1 of Example 3.1 the result t_4 hits three of four equally important preferences exactly and in several situations might have a higher noticed overall quality than t_2, which only satisfies one preference criterion.

Finally, after the computation of the qualities some aggregated quality information is computed from the obtained quality knowledge. The knowledge about the result quality regarding a base preference constructor provides presentation arguments as seen in Example 3.1. Thus, the number of presentation arguments with quality v_j is of interest. This leads to the following Definition 3.4.

Definition 3.4 Quality information in enhanced relations BMO^+ and BMO^{++}

Assume a BMO result set of the following form. The enhancements BMO^+ and BMO^{++} are defined as follows:

$BMO(A_1, \ldots, A_m)$
$BMO^+(A_1, \ldots, A_m, Q)$
$BMO^{++}(A_1, \ldots, A_m, Q, AQ_{v_1}, \ldots, AQ_{v_z}, AQ_{overall})$

BMO^+ is the extended temporary relation of the preselection (BMO) of the product search according to the search preference P with an additional attribute Q. There q $\in$ Q denotes an element of a complex nested type that describes the computed quality information for a tuple t with respect to all preferences involved according to Definition 2.4. BMO^{++} enhances BMO^+ by aggregated quality attributes. There $aq_{v_j} \in AQ_{v_j}$ denotes the frequency of the quality valuation v_j for a result tuple t $\in$ BMO regarding to any base preference, which is involved in P. $AQ_{overall}$ denotes the overall quality of a result tuple t mapped to a numerical value, which will be helpful in a technical matter later on. Of course BMO^{++} is extensible for each other imaginable aggregated attribute. *

5. Applying *"But-only" filter* conditions to the BMO results:

At this point there must be a decision which results of the BMO set indeed should be presented to the customer. With a "but-only" filter conditions can be declared regarding which results to hide from the customer.

Definition 3.5 "But-only" filter applied to query results

Let BOF specify a hard filter condition over BMO^{++}. Then the result set of the selection on the extended preselection BMO^{++} with application of the "but-only" filter BOF is declared as

$BMO^* := \sigma_{BOF}(BMO^{++})$,

where $\sigma_{condition}$ (R) denotes the well-known hard selection over the relation R. *

A condition can be stated in terms of quality, e.g. results which only have a very low overall quality should be disregarded. Another condition might concern the vendor's utility function, e.g. only show the five results with the highest profit. Or in combination, only show results with a turnover higher than 1000 €, but in the case of a profit margin higher than 55% also show these results.

6. Marking favorites via *presentation preferences*:
The decision has been made to present BMO* to the customer, but for a promising presentation the vendor should focus on one or more results. The system should proactively point out a result to the customer and possibly give reasons why this is an appropriate result. In case of a sales scenario, the vendor would act smart and according to the rules of sales psychology, personalized to each customer. There is a wide range of rules how to decide which product to present first, second and so on. Thus, to respect the presentation preferences PP, e.g. sales strategy driven selection criteria can be formulated in any way to select the appropriate result. Within PP also vendor preferences can be formulated which supports the vendor's utility function, e.g. maximizing the profit. But of course not only in sales scenarios there are preferences which results to present first. The selection for determination of the favorites with the presentation preference PP can be declared as follows.

Definition 3.6 Presentation preference over BMO*

Let PP be the presentation preference over the enhanced result set BMO*. Then

$$\text{FAV} := \sigma[\text{PP}](\text{BMO*})$$

denotes the result set of favorite query results according to the presentation preference PP. *

If FAV consists of more than one result then a random pick out of this result set is a proper choice to determine which result to present in the first row. Yet, the empty result effect never occurs, since the selection of favorites is defined as preference.

7. Result presentation and consideration of customer *feedback*:
Finally, if there is no agreement for a result there might be several reasons. On the one side, if the number of results is still too large, then a filtering within the results should be done according to the customer's feedback. The results of this selection are computed with the customer filter criterion CFC by σ_{CFC}(BMO*). Because of the very good filter effect of the BMO query model (see [KFHE01]) this will not often be necessary. On the other side, if the customer is not satisfied with this selection then according to his feedback a further preference query should be started or because of step 5 hidden results could be presented. If the customer has correctly expressed his search preferences this will not often be necessary, because he can be sure that all (but the possibly hidden) relevant results have been presented.

Considering these 7 steps of the workflow of Definition 3.1 (see Figure 3.1) step 1 and 2 were discussed in chapter 2 in detail. Step 7 depends on the application and shall not be the focus here. For step 3 in the following three sections a user intuitive linguistic model for the quality valuation of query results and intuitive and situation based approaches for the quality valuation of base as well as complex preferences are introduced. Step 4 can easily be managed by applying the aggregate functions of the used query language and therefore needs no further consideration. In sections 3.5 and 3.6 approaches for "but-only" filters and presentation preferences for a personalized and smart presentation dialog are introduced. The concluding section of this chapter shows the compatibility of the approach to further search technologies.

3.2 An intuitive linguistic model for the quality of query results

The valuation of the quality of a search result with respect to the search preferences, separately considered for each occurring base preference, is very dependant on domain and situation. E.g. what is the quality when the preferred price should be not higher than 1200 € and the offered price amounts 1249 €? This depends on several factors of the respective situation, for example on the usual range of prices in this business sector. That means, for each base preference constructor the knowledge engineer of a preference search engine has to design valuations for the quality for different situations. Especially talking about the presentation to a human being, a human does not think in numbers when talking about qualities, e.g. he does not see a preference fulfilled with 79%. People normally just recognize less than ten different states and for example just realize a preference as nearly satisfied. People think in so called linguistic variables (Definition 3.7), introduced by Zadeh ([Zad73]) in the well known fuzzy logic.

Definition 3.7 Linguistic variable

A linguistic variable is a variable where the values are linguistic terms:

Linguistic variable: NAME_OF_THE_LINGUISTIC_VARIABLE
Domain: $\{term_1, term_2, \ldots, term_n\}$

*

As an example, in Table 3.3 some linguistic terms of the linguistic variable WEATHER CONDITION are illustrated, which is just about comprehensible. On first sight it might seem that every condition is mentioned. Indeed there are more than 30 different conditions on Weather.com[11] which is hardly understood by a human. The most-used number for rating something is five, e.g. hotels are rated up to five stars, the Amazon marketplace offers five different categories for the condition of a second hand article as well as for ratings of customer review's for e.g. books.

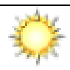
							
'sunny'	'mostly sunny'	'partly cloudy'	'mostly cloudy'	'cloudy'	'shower'	'rain'	'snow'

Table 3.3 Range of values for the linguistic variable WEATHER CONDITION

[11] http://www.weather.com

To provide an intuitive and comprehensible rating of customer preferences in comparison with a delivered search result a set of linguistic terms must be designed for the quality of a result tuple with respect to the search preference P and the underlying situation s. This linguistic model must hold for base preferences as well as for complex preferences. Of course, also this model is domain dependent, but because of classifying the quality of positive results the following model (Definition 3.8) is given, providing also five categories.

Definition 3.8 Linguistic model for the quality of a BMO search result

Linguistic variable: PREF_QUAL
Domain: (‘perfect’, ‘very good’, ‘good’, ‘acceptable’, ‘sufficient‘)

The domain for these quality valuations is defined as a descending ordered list. *

This model represents the maximum of different categories a human normally recognizes of positive quality valuations and is therefore appropriate for the valuation of search results. In this, the valuation ‘sufficient‘ denotes a condition which is far away from the customer’s preference but still a possible alternative.

With this model the quality valuation function $QUAL_{P,s}$ of Definition 3.2 is instantiated in Definition 3.9 and used in the sequel. In the next sections it becomes obvious that also different linguistic models could be used, but in this thesis the human comprehensible five categories are suggested.

Definition 3.9 Instantiated quality function $QUAL_{P,s}$

According to Definition 3.2 under usage of the linguistic model of Definition 3.8 the quality valuation function is instantiated as:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} , & t[A] \in C_1(s) \\ \text{'very good'} , & t[A] \in C_2(s) \\ \text{'good'} , & t[A] \in C_3(s) \\ \text{'acceptable'}, & t[A] \in C_4(s) \\ \text{'sufficient'} , & t[A] \in C_5(s) \end{cases}$$

*

3.3 Quality information for base preferences

In contrast to ad hoc valuations and non-intuitive approaches in the sequel a universal and semantics based approach is developed. The preference model of [Kie02, Kie04] provides semantic information according to various intuitive constructs to formulate a human’s wish. Based on this framework, in the sequel the quality of a search result is determined under consideration of this semantics. Thus, the obtained qualities are available with an according context and so they provide comprehensible, suitable arguments for the presentation of search results. In this section the qualities for the basic constructs of the described preference framework is defined well-formed for the base preferences SCORE, BETWEEN, AROUND, LOWEST, HIGHEST, AT_LEAST, AT_MOST, EXPLICIT, LAYERED_m, POS/NEG, POS/POS, NEG, POS, and ANTI-CHAIN. Categorizing the qualities of a result tuple within its queried base preferences is a very sensitive design decision and depends on many factors of the respective situation. In the sequel, the quality valuation function $\text{QUAL}_{P,s}$ of Definition 3.9 is specified in a well-formed way for each mentioned base preference.

Because of the sensitivity of the quality valuations to the current situation, partitioning parameters $b_i(s) \in \mathbb{R}$ are defined, depending on the situation s. The difficulties how to instantiate the function $\text{QUAL}_{P,s}$ for a special situation are discussed by means of accompanying examples. Please note that consequently $\text{QUAL}_{P,s}$ is instantiated for a BMO result set as defined in Definition 3.2/Definition 3.9. In section 3.7 it is shown that the following valuation approach is not restricted to the intuitive BMO model.

3.3.1 SCORE preference

Definitely a perfect match for a preference P := SCORE(A, f) is a match where the scoring function is maximal. But often the scoring function f has no upper limit in a specific domain. Thus for a quality valuation additional knowledge about the situation must be regarded. Depending on the specific situation the knowledge engineer has to formulate $\text{QUAL}_{P,s}$ as declared in Definition 3.10.

Definition 3.10 $\text{QUAL}_{P,s}$-function of a SCORE preference

For a given preference P := SCORE(A, f) in a situation s and a result tuple $t \in$ BMO, $\text{QUAL}_{P,s}$ is of the following form, where $b_1(s) \geq b_2(s) \geq b_3(s) \geq b_4(s)$:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & b_1(s) \leq f(t[A]) \\ \text{'very good'}, & b_2(s) \leq f(t[A]) < b_1(s) \\ \text{'good'}, & b_3(s) \leq f(t[A]) < b_2(s) \\ \text{'acceptable'}, & b_4(s) \leq f(t[A]) < b_3(s) \\ \text{'sufficient'}, & f(t[A]) < b_4(s) \end{cases}$$

*

Because SCORE is the top constructor within the sub-constructor hierarchy of all base preferences except the EXPLICIT preference the design of correct $\text{QUAL}_{P,s}$-functions for other base preferences can be developed from Definition 3.10 by specifying the respective f-function. Example 3.2 shows a typical use case for the SCORE preference valuation.

Example 3.2 Sample $\text{QUAL}_{P,s}$-function for a SCORE preference

Moe is a stock exchange specialist and has developed a very special chart analysis for stock quotations. As an online service provider for different investors he delivers different key data. E.g. for short-term speculative investments, he has developed the following score function for shares:

$$f(x, y, z) := x \cdot \frac{y}{z},$$

where x := relative exchange rate difference in per cent to the day before,
y := trading volume of the day,
z := the overall volume of share values of the concerning enterprise.

Table 3.4 shows a sample excerpt of a daily updated database, extended by the results of the scoring function f(x, y, z).

	enterprise	x	y	z	f(x,y,z)
t_1	Coal & Quarry	5.07	39842.40	5108000	0.0395
t_2	Alaska Beer	-3.08	369783.18	27761500	-0.0410
t_3	Zodiac Inc.	12.23	868375.20	7459200	1.4238
t_4	Nozzle Prod.	1.72	213999.12	1285280	0.2864

Table 3.4 Sample excerpt of a stock exchange database

In Moe's daily recommendation the best match(es) regarding to the SCORE preference P := SCORE((x, y, z), $x \cdot \frac{y}{z}$) are presented to the customers of this service. But of course there are more or less speculative days. For a presentation dialog to a human the online service should be able to tell about the quality of the recommendation. To design the quality function for this SCORE preference the experiences of this service provider and his customers are very important. As an expert for this domain, Moe designs the quality for his online service for his customer Homer as follows:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & 1.2 \le f(x,y,z) \\ \text{'very good'}, & 0.8 \le f(x,y,z) < 1.2 \\ \text{'good'}, & 0.6 \le f(x,y,z) < 0.8 \\ \text{'acceptable'}, & 0 \le f(x,y,z) < 0.6 \\ \text{'sufficient'}, & f(x,y,z) < 0 \end{cases}$$

Of course, for other customers or different situations Moe might develop different instances of $QUAL_{P,s}(t)$. Considering the sample trading day of Table 3.4 tuple t_3 is the best match with respect to the mentioned SCORE preference. At this day the best match is a perfect match in terms of Moe's function. He can recommend via online service the following:

> *"Good morning Homer. You are once again interested in a short-term speculative investment. Today my analysis has calculated a **perfect** occasion for this venture. I strongly recommend Zodiac Inc., very volatile with a high chance to make profit."*

Assuming a trading day with exactly the same values of Table 3.4 but without t_3, then the recommendation might be:

> *"Good morning Homer. You are once again interested in a short-term speculative investment. Today my analysis has calculated a very calm day. An **acceptable** chance to make profit is given by buying Nozzle Prod."*

In this domain, even the situation of results rated only 'sufficient' could be a gain for the service provider. Assuming a trading day where t_2 of Table 3.4 is the top result. The stock exchange specialist Moe can increase his credibility by recommending the following:

> *"Good morning Homer. In my opinion, today is a very **bad** day for a short-term speculation. My **best** share today is the **sufficient** Alaska Beer. But perhaps tomorrow I can give you a better recommendation."*

⌂

Pointing out search results via natural language generation ([RD97]) is a very advanced form that effectively uses the possibility of the developed semantic information. Obviously, also other techniques like e.g. highlighting characteristics of results gain from the quality information. Yet, for an intuitive and believable valuation, $QUAL_{P,s}$ must satisfy the postulate of Definition 3.3. Moreover, unordered results here are intuitively required to be of the same quality since unordered elements are elements of the same score.

Lemma 3.1

For the $QUAL_{P,s}$-function of the SCORE preference it holds that

a) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t <_P t' \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$.

b) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t \parallel_P t' \Rightarrow QUAL_{P,s}(t) = QUAL_{P,s}(t')$. *

Proof 3.1

a) 1.) $t <_P t'$

2.) $t <_P t' \Leftrightarrow f(t[A]) < f(t'[A])$

3.) Assumption: $\exists\, t, t'$: $t <_P t'$ and $QUAL_{P,s}(t) > QUAL_{P,s}(t')$

Then according to the definition of $QUAL_{P,s}$ $\exists\, k$: $f(t'[A]) < b_k(s) \leq f(t[A])$ which is a contradiction to 2.)

b) $t \parallel_P t' \Leftrightarrow f(t[A]) = f(t'[A]) \Rightarrow$ proposition, because of the definition of $QUAL_{P,s}(t)$. □

The full-text preferences as introduced in [LK02] deliver a score for the information retrieval. Please note that the search result quality of such a full-text preference result is compatible to the $QUAL_{P,s}$-function of the SCORE preference.

3.3.2 BETWEEN preference

Regarding a preference P := BETWEEN(A, [low, up]), a search result t is marked as a perfect match if and only if $t[A] \in [low, up]$. For the quality classifications except 'perfect' the knowledge engineer has to declare the suitable ranges, under involvement of the respective situation. There, a result with a distance d > 0 from the upper bound of the optimal range intuitively must be of the same quality like a result with distance d from the lower bound. Therefore, the following symmetric partition is defined.

Definition 3.11 $QUAL_{P,s}$-function of a BETWEEN preference

For a given preference P := BETWEEN(A, [low, up]) in a situation s for a result tuple $t \in$ BMO, $QUAL_{P,s}$ is of the following form, where $0 \le b_1(s) \le b_2(s) \le b_3(s)$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} , & low \le t[A] \le up \\ \text{'very good'} , & low - b_1(s) \le t[A] < low \vee up < t[A] \le up + b_1(s) \\ \text{'good'} , & low - b_2(s) \le t[A] < low - b_1(s) \vee up + b_1(s) < t[A] \le up + b_2(s) \\ \text{'acceptable'}, & low - b_3(s) \le t[A] < low - b_2(s) \vee up + b_2(s) < t[A] \le up + b_3(s) \\ \text{'sufficient'} , & t[A] < low - b_3(s) \vee up + b_3(s) < t[A] \end{cases}$$

*

Example 3.3 Sample $QUAL_{P,s}$-function for a BETWEEN preference

In Apu's e-procurement portal customers are interested in clear stock boxes of a special length. E.g. the craftsman Barnie is looking for a clear stock box of length between 32 cm and 35 cm, expressed by a preference P := BETWEEN(length, [32, 35]). E.g. a clear stock box with a length of 33 cm of course would be denoted as a result of perfect quality. The task for Apu is to decide which deviations from the optimal range can be denoted as which quality for a result t. One possibility to valuate the results for Barnie's wish might be to use absolute values, for example $b_1(s) = 2$, $b_2(s) = 4$, and $b_3(s) = 7$. Then the corresponding $QUAL_{P,s}$-function is the following:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} , & 32 \le length \le 35 \\ \text{'very good'} , & 30 \le length < 32 \vee 35 < length \le 37 \\ \text{'good'} , & 28 \le length < 30 \vee 37 < length \le 39 \\ \text{'acceptable'}, & 25 \le length < 28 \vee 39 < length \le 42 \\ \text{'sufficient'} , & length < 25 \vee 42 < length \end{cases}$$

A second possibility for Apu might be a design relative to the given desired range. He may have the experience that customers, who tell a wider range as preferred length, also accept higher distances from the optimum range. The difference between up and low is denoted as

$x := up - low$. A proper modeling of the $QUAL_{P,s}$-function might be for example $b_1(s) = x$, $b_2(s) = 2x$, and $b_3(s) = 3x$. In Barnie's case $x = 3$ and the $QUAL_{P,s}$-function would be the following:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & 32 \le length \le 35 \\ \text{'very good'}, & 29 \le length < 32 \vee 35 < length \le 38 \\ \text{'good'}, & 26 \le length < 29 \vee 38 < length \le 41 \\ \text{'acceptable'}, & 23 \le length < 26 \vee 41 < length \le 44 \\ \text{'sufficient'}, & length < 23 \vee 44 < length \end{cases}$$

⌂

3.3.3 AROUND preference

The AROUND preference is a special case of the BETWEEN preference and has therefore a very similar modeling for the $QUAL_{P,s}$-function by setting $up = low =: z$.

Definition 3.12 $QUAL_{P,s}$-function of an AROUND preference

For a given preference $P := AROUND(A, z)$ in a situation s for a result tuple $t \in BMO$, $QUAL_{P,s}$ is of the following form, where $0 \le b_1(s) \le b_2(s) \le b_3(s)$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & t[A] = z \\ \text{'very good'}, & z - b_1(s) \le t[A] < z \vee z < t[A] \le z + b_1(s) \\ \text{'good'}, & z - b_2(s) \le t[A] < z - b_1(s) \vee z + b_1(s) < t[A] \le z + b_2(s) \\ \text{'acceptable'}, & z - b_3(s) \le t[A] < z - b_2(s) \vee z + b_2(s) < t[A] \le z + b_3(s) \\ \text{'sufficient'}, & t[A] < z - b_3(s) \vee z + b_3(s) < t[A] \end{cases}$$

*

The only difference to the BETWEEN preference is the coincidence of the 'up' and 'low' bounds of the desired interval. Thus, a result tuple t is denoted as perfect match only when it is exactly the value z. For this special case of the BETWEEN preference once again the same domain is considered.

Example 3.4 Sample $QUAL_{P,s}$-function for an AROUND preference

In Apu's e-procurement portal some customers have a more exact imagination of the desired length of a clear stock box. E.g. Selma desires boxes with a length of around 34 cm. This means that the length of the clear stock box should be 34 cm or as a best alternative nearest to 34 cm. For this preference $P := AROUND(length, 34)$ Apu designs the $QUAL_{P,s}$-function with $b_1(s) = 2$, $b_2(s) = 4$, and $b_3(s) = 7$:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{length} = 34 \\ \text{'very good'}, & 32 \le \text{length} < 34 \vee 34 < \text{length} \le 36 \\ \text{'good'}, & 30 \le \text{length} < 32 \vee 36 < \text{length} \le 38 \\ \text{'acceptable'}, & 27 \le \text{length} < 30 \vee 38 < \text{length} \le 41 \\ \text{'sufficient'}, & \text{length} < 27 \vee 41 < \text{length} \end{cases}$$

⌂

Lemma 3.2

For the $\mathrm{QUAL}_{P,s}$-functions of the BETWEEN and the AROUND preference it holds that

a) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t <_P t' \Rightarrow \mathrm{QUAL}_{P,s}(t) \le \mathrm{QUAL}_{P,s}(t')$.

b) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t \,\|_P\, t' \Rightarrow \mathrm{QUAL}_{P,s}(t) = \mathrm{QUAL}_{P,s}(t')$. *

Proof 3.2

Following the sub-constructor hierarchy the $\mathrm{QUAL}_{P,s}$-function of the BETWEEN preference is an instance of the $\mathrm{QUAL}_{P,s}$-function of the SCORE preference with f(t[A]) := -distance(t[A], [low, up]) as follows. Let $b_{iS}(s)$ the renamed partitioning parameters $b_i(s)$ of the $\mathrm{QUAL}_{P,s}$-function of the SCORE preference. Then with parameters $b_i(s)$ for the valuation of the BETWEEN preference set $b_{1S}(s) = 0$, $b_{2S}(s) = -b_1(s)$, $b_{3S}(s) = -b_2(s)$, and $b_{4S}(s) = -b_3(s)$. Used in the $\mathrm{QUAL}_{P,s}$-function of the SCORE preference with $0 \le b_1(s) \le b_2(s) \le b_3(s)$ it reads:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & 0 \le -\text{distance}(t[A],[\text{low},\text{up}]) \\ \text{'very good'}, & -b_1(s) \le -\text{distance}(t[A],[\text{low},\text{up}]) < 0 \\ \text{'good'}, & -b_2(s) \le -\text{distance}(t[A],[\text{low},\text{up}]) < -b_1(s) \\ \text{'acceptable'}, & -b_3(s) \le -\text{distance}(t[A],[\text{low},\text{up}]) < -b_2(s) \\ \text{'sufficient'}, & -\text{distance}(t[A],[\text{low},\text{up}]) < -b_3(s) \end{cases}$$

With the properties of distance(t[A], [low, up]) it follows:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{low} \le t[A] \le \text{up} \\ \text{'very good'}, & -b_1(s) \le -\text{low} + t[A] < 0 \vee -b_1(s) \le \text{up} - t[A] < 0 \\ \text{'good'}, & -b_2(s) \le -\text{low} + t[A] < -b_1(s) \vee -b_2(s) \le \text{up} - t[A] < -b_1(s) \\ \text{'acceptable'}, & -b_3(s) \le -\text{low} + t[A] < -b_2(s) \vee -b_3(s) \le \text{up} - t[A] < -b_2(s) \\ \text{'sufficient'}, & -\text{low} + t[A] < -b_3(s) \vee \text{up} - t[A] < -b_3(s) \end{cases}$$

$\Leftrightarrow$

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{low} \le t[A] \le \text{up} \\ \text{'very good'}, & \text{low} - b_1(s) \le t[A] < \text{low} \vee \text{up} < t[A] \le \text{up} + b_1(s) \\ \text{'good'}, & \text{low} - b_2(s) \le t[A] < \text{low} - b_1(s) \vee \text{up} + b_1(s) < t[A] \le \text{up} + b_2(s) \\ \text{'acceptable'}, & \text{low} - b_3(s) \le t[A] < \text{low} - b_2(s) \vee \text{up} + b_2(s) < t[A] \le \text{up} + b_3(s) \\ \text{'sufficient'}, & t[A] < \text{low} - b_3(s) \vee \text{up} + b_3(s) < t[A] \end{cases}$$

The latter denotes the $QUAL_{P,s}$-function of the BETWEEN preference. Thus, Lemma 3.2 holds, because the $QUAL_{P,s}$-function of the SCORE function satisfies a) and b). The same holds for the $QUAL_{P,s}$-function of the AROUND preference, because it is derived from the $QUAL_{P,s}$-function of the BETWEEN preference by setting up = low =: z. □

3.3.4 LOWEST preference

In the LOWEST preference the knowledge engineer has to decide which quality valuation a search result deserves. With knowledge about the situation he can design the following well-formed $QUAL_{P,s}$-function.

Definition 3.13 $QUAL_{P,s}$-function of a LOWEST preference

For a given preference P := LOWEST(A) in a situation s for a result tuple t ∈ BMO, $QUAL_{P,s}$ is of the following form, where $b_1(s) \leq b_2(s) \leq b_3(s) \leq b_4(s)$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & t[A] \leq b_1(s) \\ \text{'very good'}, & b_1(s) < t[A] \leq b_2(s) \\ \text{'good'}, & b_2(s) < t[A] \leq b_3(s) \\ \text{'acceptable'}, & b_3(s) < t[A] \leq b_4(s) \\ \text{'sufficient'}, & b_4(s) < t[A] \end{cases}$$

*

Please note that the partition for the perfect valuation is not restricted. The reason for that becomes clear in Example 3.5.

Example 3.5 Sample $QUAL_{P,s}$-function for a LOWEST preference

Bart is a registered customer of the online audio CD shop of Apu and is always interested in cheap CDs. That means Bart has a preference P := LOWEST(price) for a CD. For a result tuple t a preference quality function might be designed by Apu as follows. He knows about Bart's price imaginations, because Bart often had complained via e-mail.

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & price \leq 4 \\ \text{'very good'}, & 4 < price \leq 6 \\ \text{'good'}, & 6 < price \leq 9 \\ \text{'acceptable'}, & 9 < price \leq 10 \\ \text{'sufficient'}, & 10 < price \end{cases}$$

Because Bart is always very short of pocket-money he has very aggressive price imaginations. When Apu does not exactly know about the customer's price imaginations he uses his domain knowledge about the general experiences with various customers that tells him that the acceptable price for audio CDs on the market for price conscious customers is between 16 and 18 €. His $QUAL_{P,s}$-function for unfamiliar customers, namely for the unknown or default case, is the following:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & price \le 10 \\ \text{'very good'}, & 10 < price \le 14 \\ \text{'good'}, & 14 < price \le 16 \\ \text{'acceptable'}, & 16 < price \le 18 \\ \text{'sufficient'}, & 18 < price \end{cases}$$

⌂

Especially in a sales scenario the price domain has a natural lower bound, i.e. 0. But of course customers have an individual imagination of what a real bargain is, namely a bound when they realize their lowest price preference perfectly fulfilled. For the partition of the perfect valuated results an open range is given instead of a fixed minimum, since the finite infimum of the real world can be lower than an individual bound. Example 3.5 also shows some variety of situations depending on the persons involved. Apu has to be aware for which kind of customer the $\mathrm{QUAL}_{P,s}$-function is adequate. E.g. with techniques of user modeling ([KF89]), Apu can create instances of $\mathrm{QUAL}_{P,s}$ for some stereotype user groups. Moreover he must be aware of the structure of his product database. This $\mathrm{QUAL}_{P,s}$-function is suitable only for audio CD's, not for DVD's etc.

3.3.5 HIGHEST preference

The dual preference to the LOWEST preference is the HIGHEST preference. Therefore the definition for the $\mathrm{QUAL}_{P,s}$-function is just dual.

Definition 3.14 $\mathrm{QUAL}_{P,s}$-function of a HIGHEST preference

For a given preference P := HIGHEST(A) in a situation s for a result tuple t ∈ BMO, $\mathrm{QUAL}_{P,s}$ is of the following form, where $b_1(s) \ge b_2(s) \ge b_3(s) \ge b_4(s)$:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & b_1(s) \le t[A] \\ \text{'very good'}, & b_2(s) \le t[A] < b_1(s) \\ \text{'good'}, & b_3(s) \le t[A] < b_2(s) \\ \text{'acceptable'}, & b_4(s) \le t[A] < b_3(s) \\ \text{'sufficient'}, & t[A] < b_4(s) \end{cases}$$

*

The following example also considers an online shop scenario.

Example 3.6 Sample $\mathrm{QUAL}_{P,s}$-function for a HIGHEST preference

Apu also sells computer hardware. A fast CPU is a very important factor for a PC. For online customers with a preference for high speed engines the preference P := HIGHEST(clock frequency) can be constructed. The quality of a search result tuple t can be modeled as follows:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} , & 3.0 \le \text{clock frequency} \\ \text{'very good'} , & 2.5 \le \text{clock frequency} < 3.0 \\ \text{'good'} , & 2.3 \le \text{clock frequency} < 2.5 \\ \text{'acceptable'}, & 2.0 \le \text{clock frequency} < 2.3 \\ \text{'sufficient'} , & \text{clock frequency} < 2.0 \end{cases}$$

⌂

Example 3.6 also shows an update problem. At the moment where a new and faster CPU is introduced into market, the bound of the partition in this function must be updated. An automated update would be a proper solution. E.g. an information agent might be able to inform a shop about the clock frequency of the latest CPU.

Lemma 3.3

For the $\text{QUAL}_{P,s}$-functions of the HIGHEST and the LOWEST preference it holds that

a) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t <_P t' \Rightarrow \text{QUAL}_{P,s}(t) \le \text{QUAL}_{P,s}(t')$.

b) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t \,||_P\, t' \Rightarrow \text{QUAL}_{P,s}(t) = \text{QUAL}_{P,s}(t')$. *

Proof 3.3

The $\text{QUAL}_{P,s}$-function of the SCORE preference satisfies a) and b). The $\text{QUAL}_{P,s}$-functions of the HIGHEST and the LOWEST preferences are instances of the $\text{QUAL}_{P,s}$-function of the SCORE preference with f(t[A]) = t[A] and f(t[A]) = -t[A], respectively. For the $\text{QUAL}_{P,s}$-function of the HIGHEST preference the partitioning parameters $b_i(s)$ are exactly the one of the SCORE preference.

For $\text{QUAL}_{P,s}$-function of the LOWEST preference the partitioning parameters are the negated parameters of the SCORE preference. Used in the $\text{QUAL}_{P,s}$-function of the SCORE preference with $b_1(s) \le b_2(s) \le b_3(s) \le b_4(s)$ it reads:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} , & - b_1(s) \le - t[A] \\ \text{'very good'} , & - b_2(s) \le - t[A] < -b_1(s) \\ \text{'good'} , & - b_3(s) \le - t[A] < -b_2(s) \\ \text{'acceptable'}, & - b_4(s) \le - t[A] < -b_3(s) \\ \text{'sufficient'} , & - t[A] < -b_1(s) \end{cases}$$

$\Leftrightarrow$

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} , & t[A] \le b_1(s) \\ \text{'very good'} , & b_1(s) < t[A] \le b_2(s) \\ \text{'good'} , & b_2(s) < t[A] \le b_3(s) \\ \text{'acceptable'}, & b_3(s) < t[A] \le b_4(s) \\ \text{'sufficient'} , & b_4(s) < t[A] \end{cases}$$

The latter denotes the $\text{QUAL}_{P,s}$-function of the LOWEST preference □

Please note that, as discussed above, there possibly is a difference between the personalized upper/lower bound characterizing a perfect result and the supremum/infimum of the real world. So, the $QUAL_{P,s}$-functions of the HIGHEST/LOWEST preference cannot be instantiated from their immediate parent constructor's $QUAL_{P,s}$-function in general, but from the SCORE constructor.

3.3.6 AT_LEAST preference

From the BETWEEN preference the AT_LEAST preference P := AT_LEAST(A, z) can be derived by setting the upper bound to the supremum of the real world.

Definition 3.15 $QUAL_{P,s}$-function of an AT_LEAST preference

For a given preference P := AT_LEAST(A, z) in a situation s for a result tuple $t \in$ BMO, $QUAL_{P,s}$ is of the following form, where $0 \leq b_1(s) \leq b_2(s) \leq b_3(s)$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & z \leq t[A] \\ \text{'very good'}, & z - b_1(s) \leq t[A] < z \\ \text{'good'}, & z - b_2(s) \leq t[A] < z - b_1(s) \\ \text{'acceptable'}, & z - b_3(s) \leq t[A] < z - b_2(s) \\ \text{'sufficient'}, & t[A] < z - b_3(s) \end{cases}$$

*

The quality function of the AT_LEAST preference is technically equivalent to the $QUAL_{P,s}$-function of the HIGHEST preference (see Definition 3.14), but with a given first bound by the parameter z.

Example 3.7 Sample $QUAL_{P,s}$-function for an AT_LEAST preference

When Marge is searching in the online restaurant guide for a family dinner she prefers restaurants with at least 20 different fish dishes, i.e. P := AT_LEAST(number fish dishes, 20). A personalized $QUAL_{P,s}$-function for Marge could be of the following form:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & 20 \leq \text{number fish dishes} \\ \text{'very good'}, & 17 \leq \text{number fish dishes} < 20 \\ \text{'good'}, & 10 \leq \text{number fish dishes} < 17 \\ \text{'acceptable'}, & 3 \leq \text{number fish dishes} < 10 \\ \text{'sufficient'}, & \text{number fish dishes} < 3 \end{cases}$$

⌂

3.3.7 AT_MOST preference

The AT_MOST preference P := AT_MOST(A, z) is dual to the AT_LEAST preference and defined analogously. The $QUAL_{P,s}$-function of the AT_MOST preference is technically equivalent to the quality function of the LOWEST preference (see Definition 3.13), but with a given first bound by the parameter z.

Definition 3.16 $QUAL_{P,s}$-function of an AT_MOST preference

For a given preference P := AT_MOST(A, z) in a situation s for a result tuple t $\in$ BMO, $QUAL_{P,s}$ is of the following form, where $0 \le b_1(s) \le b_2(s) \le b_3(s)$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & t[A] \le z \\ \text{'very good'}, & z < t[A] \le z + b_1(s) \\ \text{'good'}, & z + b_1(s) < t[A] \le z + b_2(s) \\ \text{'acceptable'}, & z + b_2(s) < t[A] \le z + b_3(s) \\ \text{'sufficient'}, & z + b_3(s) < t[A] \end{cases}$$

*

Example 3.8 Sample $QUAL_{P,s}$-function for an AT_MOST preference

Ned plans holidays for his family. On the web sites of the Springfield travel agency he looks for destinations with at most 20 degrees Celsius, because his kids Tot and Rot are very sensible to hot weather. The $QUAL_{P,s}$-function for this preference P := AT_MOST(temperature, 20) might be designed as follows:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{temperature} \le 20 \\ \text{'very good'}, & 20 < \text{temperature} \le 21 \\ \text{'good'}, & 21 < \text{temperature} \le 23 \\ \text{'acceptable'}, & 23 < \text{temperature} \le 25 \\ \text{'sufficient'}, & 25 < \text{temperature} \end{cases}$$

⌂

Lemma 3.4

For the $QUAL_{P,s}$-functions of the AT_LEAST and the AT_MOST preference it holds that

a) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t <_P t' \Rightarrow QUAL_{P,s}(t) \le QUAL_{P,s}(t')$.

b) for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t \,||_P\, t' \Rightarrow QUAL_{P,s}(t) = QUAL_{P,s}(t')$. *

Proof 3.4

The $QUAL_{P,s}$-function of the BETWEEN preference satisfies a) and b). The $QUAL_{P,s}$-functions of the AT_LEAST and AT_MOST preference are special cases of the $QUAL_{P,s}$-function of the BETWEEN preference with a fixed upper/lower bound to the supremum/infimum of the real world. □

3.3.8 EXPLICIT preference

The EXPLICIT preference is the most extensive categorizing base preference. The quality valuation of a result tuple t with respect to an EXPLICIT preference P := EXP(A, E-graph) is by definition 'perfect' if level(t[A]) = 1, because elements of that level are maximal values. Let level k be the last level of the E-graph. All values not mentioned when expressing an EXPLICIT preference are by definition of level k+1, that is to say the level after the last level of the E-graph. These last alternatives must be stated as ‘sufficient’. Additionally, the knowledge engineer has to decide which quality values are the fitting ones for results of the levels 2 up to k. A result of a worse level must not have a better quality than a result of a better level. Please note that for the EXPLICIT preference order pairs are specified. That means, either the corresponding e-graph is at least of depth k = 2, or it is of depth k = 0, which denotes the special case of an ANTI-CHAIN preference, considered separately in this work.

Definition 3.17 $QUAL_{P,s}$-function of an EXPLICIT preference

For a given preference P := EXP(A, E-graph) with $k \geq 2$ as the depth of E-graph in a situation s for a result tuple $t \in$ BMO, $QUAL_{P,s}$ is of the following form, where $1 \leq b_1(s) \leq b_2(s) \leq k$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & level(t[A]) = 1 \\ \text{'very good'}, & 1 < level(t[A]) \leq b_1(s) \\ \text{'good'}, & b_1(s) < level(t[A]) \leq b_2(s) \\ \text{'acceptable'}, & b_2(s) < level(t[A]) \leq k \\ \text{'sufficient'}, & level(t[A]) = k+1 \end{cases}$$

*

A small use case example points out the usage of this $QUAL_{P,s}$-function.

Example 3.9 Sample $QUAL_{P,s}$-function for an EXPLICIT preference

Apu runs a new online shop for resellers where he offers a very comfortable and detailed user interface for expressing the preferences about the make of notebooks. A customer can express which brand he likes more than a different one. E.g. Marge expresses that she **likes** HP **more than** IBM. Moreover, she **likes** IBM **more than** Asus and also **more than** Sony. Sony, Toshiba, and Asus are **more desired than** Dell. Samsung **is less desired** than Dell, but Samsung is **preferred over** JVC. There is no statement about the make Elitegroup. In preference algebra Marge’s wish can be declared as follows:

P := EXP(make, {(JVC, Samsung), (Samsung, Dell), (Dell, Asus), (Dell, Sony),
(Dell, Toshiba), (Sony, IBM), (Asus, IBM), (IBM, HP)}).

The associated E-graph including the non-mentioned manufacturer Elitegroup is shown in Figure 3.2. A possible design for the preference quality function could be the following with $b_1(s) = 2$ and $b_2(s) = 4$.

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & make \in \{Toshiba, HP\} \\ \text{'very good'}, & make \in \{IBM\} \\ \text{'good'}, & make \in \{Sony, Asus, Dell\} \\ \text{'acceptable'}, & make \in \{Samsung, JVC\} \\ \text{'sufficient'}, & make \in \{Elitegroup\} \end{cases}$$

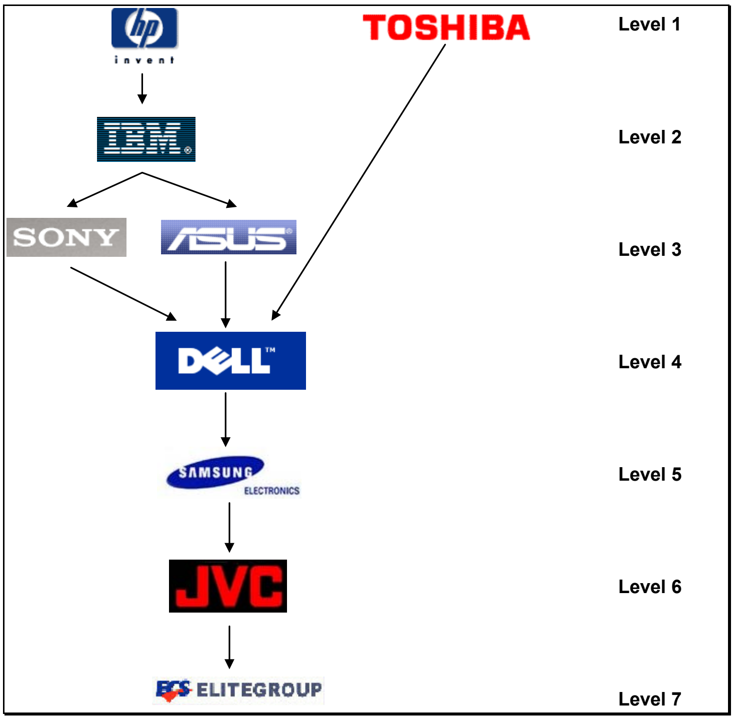


Figure 3.2 Sample E-graph for an EXPLICIT preference

In this design a make of level 3 or 4 is declared as ‘good’ quality. A make of level 5 is considered as ‘acceptable’ as a make of level 6. Assuming a more pessimistic situation following the rules of sales psychology an adequate design could be expressed by the following $QUAL_{P,s}$-function with $b_1(s) = 1.5$ and $b_2(s) = 2$:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} \;, & make \in \{Toshiba, HP\} \\ \text{'very good'} \;, & \varnothing \\ \text{'good'} \;, & make \in \{IBM\} \\ \text{'acceptable'}, & make \in \{Sony, Asus, Dell, Samsung, JVC\} \\ \text{'sufficient'} \;, & make \in \{Elitegroup\} \end{cases}$$

Because of his experience the vendor has decided not to use the quality term ‘very good’ in this case. Elements of level 2 are assigned to the ‘good’ quality term and all other mentioned values are marked as ‘acceptable’ results. ⌂

Regarding the EXPLICIT preference, unordered result tuples need not be of the same result quality, because unordered elements can be of completely different levels as discussed before. The postulate of Definition 3.3 of course must be hold.

Lemma 3.5

For the $QUAL_{P,s}$-function of the EXPLICIT preference it holds that

$$\text{for all elements } t, t' \text{ with } t[A], t'[A] \in dom(A)\text{: } t <_P t' \Rightarrow QUAL_{P,s}(t) \le QUAL_{P,s}(t'). \qquad *$$

Proof 3.5

$t <_P t' \Rightarrow \text{level}(t[A]) > \text{level}(t'[A])$. Definition 3.17 valuates elements of a higher level equal or worse. □

3.3.9 LAYERED$_m$ preference

For the preference P := LAYERED$_m$(A, L) the quality for a search result is determined by means of the according layer. Obviously, if $t[A] \in L_1$-set then it is to be declared as 'perfect'. If $t[A] \in L_{m+1}$-set, the least desired alternative, then the result tuple t has to be valuated as 'sufficient'. Similarly to the EXPLICIT preference, there is some degree of freedom how to valuate the values in between. The quality valuation for the special case m = 0 which denotes the ANTI-CHAIN preference is discussed later.

Definition 3.18 QUAL$_{P,s}$-function of a LAYERED$_m$ preference

For a given preference P := LAYERED$_m$(A, L) with $m \geq 1$ in a situation s for a result tuple $t \in$ BMO, QUAL$_{P,s}$ is of the following form, where $1 \leq b_1(s) \leq b_2(s) \leq m$:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{layer}(t[A]) = 1 \\ \text{'very good'}, & 1 < \text{layer}(t[A]) \leq b_1(s) \\ \text{'good'}, & b_1(s) < \text{layer}(t[A]) \leq b_2(s) \\ \text{'acceptable'}, & b_2(s) < \text{layer}(t[A]) \leq m \\ \text{'sufficient'}, & \text{layer}(t[A]) = m+1 \end{cases}$$

*

Example 3.10 Sample QUAL$_{P,s}$-function for a LAYERED$_m$ preference

In his new online shop for resellers Apu offers a further way for expressing the preferences about the make of notebooks. A customer can name sets of makes which he would imagine as an alternative. E.g. Marge tells now that she **prefers** HP and Toshiba. As a **first alternative** she would **like** IBM. If this is not available Sony, Dell, or Asus would be her **next choice**. If this would also be not available **then** any other make **would fit**, but the choice should be **avoiding** Elitegroup. There is no statement about the makes JVC and Samsung. In preference algebra Marge's wish can be expressed as follows:

P := LAYERED$_4$(make, ({HP, Toshiba}, {IBM}, {Sony, Dell, Asus},
'other values', {Elitegroup}).

The associated 'better-than' graph including the non-mentioned manufacturers Samsung and JVC is shown in Figure 3.3. A possible design for the preference quality function could be the following with $b_1(s) = 2$ and $b_2(s) = 3$.

$$
QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} , & make \in \{Toshiba, HP\} \\ \text{'very good'} , & make \in \{IBM\} \\ \text{'good'} , & make \in \{Sony, Asus, Dell\} \\ \text{'acceptable'}, & make \in \{Samsung, JVC\} \\ \text{'sufficient'} , & make \in \{Elitegroup\} \end{cases}
$$

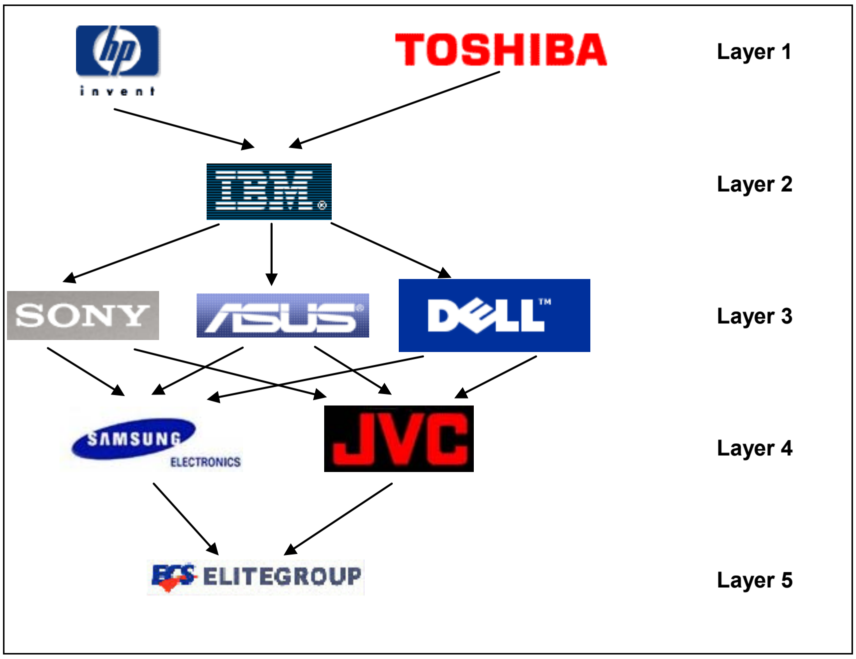


Figure 3.3 'Better-than' graph for a sample LAYERED$_m$ preference

Of course, also approaches with empty partitions are possible. ⌂

3.3.10 POS/NEG preference

POS/NEG is a sub-constructor of the LAYERED$_m$ constructor with m = 2. A result tuple t is a perfect match referring to a preference P := POS/NEG(A, POS-set; NEG-set) if t[A] ∈ POS-set and it must therefore be marked as being of 'perfect' quality. Accordingly if t[A] ∈ NEG-set, then this result t has to be declared of 'sufficient' quality. The knowledge engineer's only decision is how to valuate results t[A] ∉ POS-set ∪ NEG-set. There are the following three possibilities of designing the QUAL$_{P,s}$-function.

Definition 3.19 QUAL$_{P,s}$-function of a POS/NEG preference

For a given preference P := POS/NEG(A, POS-set; NEG-set) in a situation s for a result tuple t ∈ BMO, QUAL$_{P,s}$ is of the following form:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} & , \quad t[A] \in \text{POS-set} \\ \text{'very good'} & , \quad M \\ \text{'good'} & , \quad N \\ \text{'acceptable'} & , \quad O \\ \text{'sufficient'} & , \quad t[A] \in \text{NEG-set} \end{cases}$$

There are three possible approaches:

1. Optimistic valuation: $M := t[A] \notin$ POS-set $\cup$ NEG-set, $N := \varnothing$, $O := \varnothing$.
2. Moderate valuation: $N := t[A] \notin$ POS-set $\cup$ NEG-set, $M := \varnothing$, $O := \varnothing$.
3. Pessimistic valuation: $O := t[A] \notin$ POS-set $\cup$ NEG-set, $M := \varnothing$, $N := \varnothing$. *

It depends on the situation which approach a knowledge engineer selects. It is important in terms of sales psychology. With a more pessimistic valuation perhaps for some customers the presentation is more believable. For other customers a more offensive presentation with an optimistic valuation leads to a successful deal.

Example 3.11 Sample $\mathrm{QUAL}_{P,s}$-function for a POS/NEG preference

Marge is interested to order a bundle of notebooks from her vendor Apu to run a marketing action. She prefers notebooks from Toshiba or Hewlett Packard but has made very bad experiences with Sony laptops. Thus, Marge has a preference P := POS/NEG(make, {Toshiba, HP}; {Sony}). Apu could design a $\mathrm{QUAL}_{P,s}$-function for his online shop as follows:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} & , \quad \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'very good'} & , \quad \varnothing \\ \text{'good'} & , \quad \text{make} \notin \{\text{Toshiba, HP, Sony}\} \\ \text{'acceptable'} & , \quad \varnothing \\ \text{'sufficient'} & , \quad \text{make} \in \{\text{Sony}\} \end{cases}$$

In the following, empty parts of the partition of Definition 3.9 and the related linguistic quality terms are not mentioned. Apu only has to design how to declare results which are neither in the POS-set nor in the NEG-set. A moderate approach is to declare such results as results of good quality as seen above. A vendor with a more careful or pessimistic attitude in his result presentation can state the quality of these results only as ‘acceptable’, because he has the experience that his customers or a part of them realize a pessimistic valuation as more believable:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} & , \quad \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'acceptable'} & , \quad \text{make} \notin \{\text{Toshiba, HP, Sony}\} \\ \text{'sufficient'} & , \quad \text{make} \in \{\text{Sony}\} \end{cases}$$

Analogously, for the dual situation an optimistic valuation would be to declare these results as ‘very good’:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} &, \quad \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'very good'}, & \quad \text{make} \notin \{\text{Toshiba, HP, Sony}\} \\ \text{'sufficient'} &, \quad \text{make} \in \{\text{Sony}\} \end{cases}$$

⌂

3.3.11 POS/POS preference

The POS/POS preference is a special case of the EXPLICIT preference and the LAYERED_m preference with m = 2. Similar to the POS/NEG preference, for a POS/POS preference P := (A, POS_1-set; POS_2-set) there is only one degree of freedom in terms of the quality valuation for a tuple t $\in$ BMO. A tuple t with t[A] $\in$ POS_1-set is to be declared as 'perfect', because in this case t is a perfect match. If t[A] $\notin$ POS_1-set and t[A] $\notin$ POS_2-set then this t[A] is not mentioned by the customer and therefore the quality valuation must be 'sufficient'. Only for values t with t[A] $\in$ POS_2-set the knowledge engineer has to decide which quality valuation to use. In this, three approaches are consistent with the preference philosophy:

Definition 3.20 $\text{QUAL}_{P,s}$-function of a POS/POS preference

For a given preference P := POS/POS(A, POS_1-set; POS_2-set) in a situation s for a result tuple t $\in$ BMO, $\text{QUAL}_{P,s}$ is of the following form:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} &, \quad t[A] \in \text{POS}_1\text{-set} \\ \text{'very good'} &, \quad M \\ \text{'good'} &, \quad N \\ \text{'acceptable'} &, \quad O \\ \text{'sufficient'} &, \quad t[A] \notin \text{POS}_1\text{-set} \cup \text{POS}_2\text{-set} \end{cases}$$

There are three possible approaches:

1. Optimistic valuation: M := t[A] $\in$ POS_2-set, N := $\emptyset$, O := $\emptyset$.
2. Moderate valuation: N := t[A] $\in$ POS_2-set, M := $\emptyset$, O := $\emptyset$.
3. Pessimistic valuation: O := t[A] $\in$ POS_2-set, M := $\emptyset$, N := $\emptyset$. *

Example 3.12 Sample $\text{QUAL}_{P,s}$-function for a POS/POS preference

Once again Marge is interested in a bundle of notebooks and has a preference regarding the make. She prefers HP and Toshiba. If this is not available she likes Sony, Dell or Asus. A possible modeling of the $\text{QUAL}_{P,s}$-function is the following:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} &, \quad \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'good'} &, \quad \text{make} \in \{\text{Sony, Dell, Asus}\} \\ \text{'sufficient'}, & \quad \text{make} \notin \{\text{Sony, Dell, Asus, Toshiba, HP}\} \end{cases}$$

The quality valuation of the preferred set is obviously perfect and also the non-mentioned makes belong to 'sufficient'. The design question is how to valuate the quality of the alternative positive set ({Sony, Dell, Asus}). The design mentioned above shows the moderate approach regarding the quality. For a very careful and pessimistic approach the following definition is imaginable as well:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'acceptable'}, & \text{make} \in \{\text{Sony, Dell, Asus}\} \\ \text{'sufficient'}, & \text{make} \notin \{\text{Sony, Dell, Asus, Toshiba, HP}\} \end{cases}$$

A more optimistic approach uses the possibility to valuate the alternative positive set as 'very good' as is mentioned below:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{make} \in \{\text{Toshiba, HP}\} \\ \text{'very good'}, & \text{make} \in \{\text{Sony, Dell, Asus}\} \\ \text{'sufficient'}, & \text{make} \notin \{\text{Sony, Dell, Asus, Toshiba, HP}\} \end{cases}$$

⌂

3.3.12 NEG preference

NEG is a sub-constructor of $\mathrm{LAYERED}_m$ with m = 1. The quality valuation for a result t ∈ BMO with respect to a preference P := NEG(A, NEG-set) is intuitively clear and leaves no scope for a knowledge engineer. A result t has to be declared of 'perfect' quality when t[A] ∉ NEG-set, because that is exactly what the customer desires. If t[A] ∈ NEG-set this result must be stated as 'sufficient', according to the preference model philosophy.

Definition 3.21 $\mathrm{QUAL}_{P,s}$-function of a NEG preference

For a given preference P := NEG(A, NEG-set) in a situation s for a result tuple t ∈ BMO, $\mathrm{QUAL}_{P,s}$ is of the following form:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & t[A] \notin \text{NEG-set} \\ \text{'sufficient'}, & t[A] \in \text{NEG-set} \end{cases}$$

*

Example 3.13 Sample $\mathrm{QUAL}_{P,s}$-function for a NEG preference

Marge is interested in a bundle of notebooks from her vendor Apu to run a marketing action. Marge does not care much about the make except that she does not like notebooks from Sony, because this make was a slow seller in the last marketing action. Expressed in preference algebra, Marge's preference reads P := NEG(make, {Sony}). The vendor's correct design of a $\mathrm{QUAL}_{P,s}$-function regarding the make for his online shop is the following:

$$\mathrm{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{make} \notin \{\text{Sony}\} \\ \text{'sufficient'}, & \text{make} \in \{\text{Sony}\} \end{cases}$$

If Apu is able to deliver other notebooks than Sony, then it is a perfect match for Marge, since she prefers any kind of notebook over Sony notebooks. Otherwise if the only deliverable notebooks are from Sony, the result quality is just the term ‘sufficient‘, because she wants to avoid this make. ⌂

3.3.13 POS preference

The dual of the NEG preference is the POS preference. Thus, the design modeling is intuitively the dual design of the NEG preference. A result tuple t must be valuated of ‘perfect’ quality if t[A] ∈ POS-set regarding to a preference P := POS(A, POS-set), because that is the customer’s explicit wish. Otherwise if this wish could not be satisfied the result tuple intuitively has to be declared as ‘sufficient’.

Definition 3.22 $QUAL_{P,s}$-function of a POS preference

For a given preference P := POS(A, POS-set) in a situation s for a result tuple t ∈ BMO, $QUAL_{P,s}$ is of the following form:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} \;, & t[A] \in \text{POS-set} \\ \text{'sufficient'}, & t[A] \notin \text{POS-set} \end{cases}$$

*

Example 3.14 Sample $QUAL_{P,s}$-function for a POS preference

Marge wants to run a marketing action. Therefore she asks her vendor Apu for a notebook offer. Marge tells him that Toshiba and IBM are the preferred makes. In formal words she has a preference P := POS(make, {Toshiba, IBM}). Dual to the NEG preference the assigned $QUAL_{P,s}$-function must be designed as mentioned below:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'} \;, & \text{make} \in \{\text{Toshiba, IBM}\} \\ \text{'sufficient'}, & \text{make} \notin \{\text{Toshiba, IBM}\} \end{cases}$$

Notebooks made by Toshiba or IBM are perfect matches, because it is exactly what Marge likes. Any other result intuitively belongs to the category ‘sufficient‘. ⌂

Finally, the following Lemma 3.6 shows the correct design of the $QUAL_{P,s}$-functions of the last five categorizing base preferences. In contrast to the EXPLICIT preference unordered values must be of equal quality, because the unordered values are arranged in sets of the same level. E.g. it would not be understood by a customer if a green car gets a different quality rating than a red car if green and red are the colors of his POS-set in his preference POS(color, {green, red}).

Lemma 3.6

For the $QUAL_{P,s}$-functions of the categorical preferences $LAYERED_m$, POS/NEG, POS/POS, POS, and NEG it holds that

a) for all elements t, t’ with t[A], t’[A] ∈ dom(A): $t <_P t' \Rightarrow QUAL_{P,s}(t) \le QUAL_{P,s}(t')$.

b) for all elements t, t' with t[A], t'[A] $\in$ dom(A): t $\|_P$ t' $\Rightarrow$ $QUAL_{P,s}$(t) = $QUAL_{P,s}$(t'). *

Proof 3.6

Both parts of the Lemma are obvious because of the definitions of the $QUAL_{P,s}$-functions for the respective preferences.

a) t $<_P$ t' $\Rightarrow$ level(t[A]) > level(t'[A]). The definitions of the respective $QUAL_{P,s}$-functions valuate elements of a higher level equal or worse.

b) If t and t' are unordered then they are of the same level. All elements of one level are according to the partitioning of $QUAL_{P,s}$ valuated of same quality.

Alternatively, this proof can be done by using the sub-constructor hierarchy property and creating an instance of the $QUAL_{P,s}$-function of the SCORE preference for the $LAYERED_m$ preference. The other preferences are special cases of the $LAYERED_m$ preference as mentioned. The scoring function is denoted with f(t[A]) := -layer(t[A]). Let $b_{iS}(s)$ the renamed partitioning parameters of the $QUAL_{P,s}$-function of the SCORE preference. Then with parameters $b_i(s)$ for the valuation of the $LAYERED_m$ preference set $b_{1S}(s) = -1$, $b_{2S}(s) = -b_1(s)$, $b_{3S}(s) = -b_2(s)$, and $b_{4S}(s) = -m$. Used in the $QUAL_{P,s}$-function of the SCORE preference with $1 \le b_1(s) \le b_2(s) \le m$ it reads:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & -1 \le -\text{layer}(t[A]) \\ \text{'very good'}, & -b_1(s) \le -\text{layer}(t[A]) < -1 \\ \text{'good'}, & -b_2(s) \le -\text{layer}(t[A]) < -b_1(s) \\ \text{'acceptable'}, & -m \le -\text{layer}(t[A]) < -b_2(s) \\ \text{'sufficient'}, & -\text{layer}(t[A]) < -m \end{cases}$$

$\Rightarrow$

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & \text{layer}(t[A]) = 1 \\ \text{'very good'}, & 1 < \text{layer}(t[A]) \le b_1(s) \\ \text{'good'}, & b_1(s) < \text{layer}(t[A]) \le b_2(s) \\ \text{'acceptable'}, & b_2(s) < \text{layer}(t[A]) \le m \\ \text{'sufficient'}, & \text{layer}(t[A]) = m+1 \end{cases}$$

The latter denotes the $QUAL_{P,s}$-function of the $LAYERED_m$ preference. □

3.3.14 ANTI-CHAIN preference

$A^{\leftrightarrow}$, the so called ANTI-CHAIN preference is a sub-constructor of the $LAYERED_m$ constructor with m = 0. In an ANTI-CHAIN preference no value is preferred over another. E.g. a customer is interested in a new car and does not care about the color. With this short example it becomes clear that, because the customer does not matter the color, there must not be a quality valuation. That means, $QUAL_{P,s}$ is not defined for the ANTI-CHAIN preference.

With the quality valuations of each search result referring to each base preference the knowledge engineer of a preference search engine has lots of single comprehensible arguments for a qualitative result presentation. For the opening of a result presentation it is often appropriate to give an overall impression of the presented search result.

3.4 Quality information for complex preferences

When presenting one or more search results a cooperative system can improve the communication by telling the customer, how good is/are the search result(s) in general. With respect to the search preferences this means that the system must be able to qualify complex customer preferences. Thus, an overall impression for the qualities of the results would be available. Naturally, the linguistic quality terms for complex preferences are the same as for base preferences (see Definition 3.8).

Obviously, the quality of a complex preference depends on the qualities of the preferences combined within this complex preference. For computing a quality valuation for complex preferences different approaches are imaginable, e.g. a more optimistic or pessimistic variant or a moderate valuation. Which approach is the appropriate one depends on the situation. Within the result set of a preference search with complex preferences the subjective overall impressions can differ for the different result tuples t. As seen in Example 3.1 some of the unordered results hit more of the single characteristics with respect to the search preferences than the others. Yet, for a believable and intuitive presentation the postulate of Definition 3.3 must hold. This means that less preferred elements must not be valuated better than more preferred ones. In the sequel, definitions of intuitive $QUAL_{P,S}$-functions are introduced for nested complex preferences. Also negative and unintuitive designs are discussed. The approaches are independently designed for prioritized, numerical, and Pareto preferences and can therefore recursively be used for nested preferences.

3.4.1 Quality valuation of a Pareto preference

In designing the quality of a complex preference several factors must be considered wrt. the situation in which the preference search engine is used, e.g. the underlying domain and the customer's reaction depending on his personality. For example, several customers react positive to a more optimistic overall valuation of a result, i.e. to a more offensive presentation of goods. In contrast, for a more skeptical and careful customer a pessimistic overall valuation provides a more reliable effect. In the sequel two $QUAL_{P,S}$-functions for these two intuitive but extreme approaches are specified. Please note that for a Pareto preference it is assumed that no of the combined preferences is a Pareto preference itself. E.g. a preference $P := P_1 \otimes P_2$ with $P_2 := P_3 \otimes P_4$ is considered to be in the form $P := P_1 \otimes P_3 \otimes P_4$. This reflects that each of the equally important preferences P_1, P_3, and P_4 must have the same impact for the quality valuation. Obviously, in a numerical average valuation the first consideration would lead to a wrong quality.

Definition 3.23 Optimistic quality valuation of a Pareto preference result

Assume a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, in a situation s where for all P_j, $j = 1, \ldots, d$, it holds that P_j is no Pareto preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The optimistic valuation for a result tuple $t \in BMO$ is defined as follows regarding the order of quality terms of Definition 3.8:

$QUAL_{P,s}(t) := \max\{\{QUAL_{P_j,s}(t) \mid j = 1, \ldots, d\}\}$, where $\{\{\ldots\}\}$ denotes a multi-set. *

Thus, for the optimistic valuation the best quality valuation of all Pareto combined preferences P_j within P is decisive. Contrary to this approach, in the pessimistic quality valuation the worst quality valuation of the combined preferences is decisive.

Definition 3.24 Pessimistic quality valuation of a Pareto preference result

Assume a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, in a situation s where for all P_j, $j = 1, \ldots, d$, it holds that P_j is no Pareto preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The pessimistic valuation for a result tuple $t \in BMO$ is defined as follows regarding the order of quality terms of Definition 3.8:

$QUAL_{P,s}(t) := \min\{\{QUAL_{P_j,s}(t) \mid j = 1, \ldots, d\}\}$, where $\{\{\ldots\}\}$ denotes a multi-set. *

Next, these two quality valuations are illustrated in an example.

Example 3.15 Sample optimistic and pessimistic quality valuation

Marge and Homer are interested in a bundle of notebooks and consider seven equally important characteristics. Formulated in preference algebra they have the Pareto preference

$$P := P_{make} \otimes P_{mhz} \otimes P_{ram} \otimes P_{oquan} \otimes P_{del\ time} \otimes P_{weight} \otimes P_{price}.$$

For these seven base preferences Marge and Homer have the same situational parameters and the same opinions, whether these characteristics are suitably fulfilled or not. In Table 3.5 the preference quality information for each P_j is given for one result tuple t regarding to the notebook characteristics. In this, five stars for one P_j denote a ‘perfect’ quality, four stars denote a ‘very good' quality, and so on until one star denotes a ‘sufficient’ quality for the $QUAL_{P_j,s}$-function. Thus, e.g. $QUAL_{P_{make},s}(t)$ = ‘very good’. Homer is the carefree customer and only considers the positive aspects of a search result. Therefore, for Homer the optimistic quality valuation would be appropriate. With the valuation of Definition 3.23 the quality for P is computed as follows:

$QUAL_{P,s}(t)$ = max{{‘very good’, ‘acceptable’, ‘very good’,
‘very good’, ‘very good’, ‘acceptable’, ‘good’}} = ’very good’.

Marge, in contrast to Homer, is always very skeptical. For her, the pessimistic quality valuation might be suitable. With Definition 3.24 it is calculated as follows:

$$QUAL_{P,s}(t) = \min\{\{\text{'very good', 'acceptable', 'very good', 'very good', 'very good', 'acceptable', 'good'}\}\} = \text{'acceptable'}.$$

The qualities for these two examples are visualized in Table 3.5.

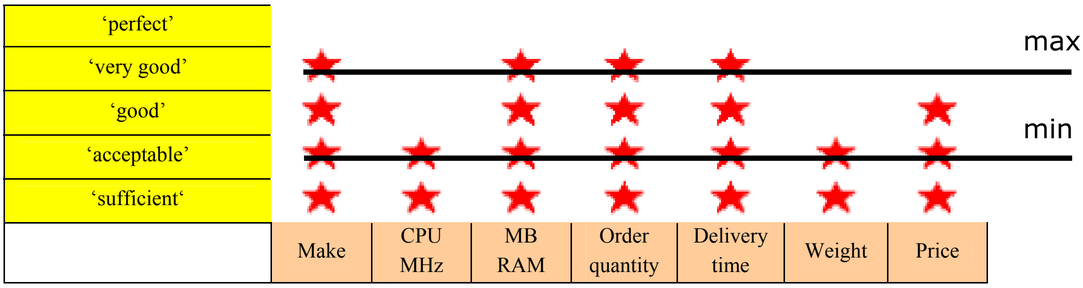


Table 3.5 Sample qualities of notebook characteristics for one result tuple t

⌂

Lemma 3.7

For the optimistic and pessimistic quality valuation of a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, it holds that

for all elements t, t' with $t[A], t'[A] \in dom(A)$: $t <_P t' \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$. *

Proof 3.7

All preferences P_j satisfy the postulate of Definition 3.3. Thus, by $t <_P t'$ and the definition of a Pareto preference it holds that

for all j = 1, …, d: $QUAL_{P_j,s}(t) \leq QUAL_{P_j,s}(t')$

$\Rightarrow$

1. for the optimistic quality valuation:
   $\max\{\{QUAL_{P_j,s}(t) \mid j = 1, \ldots, d\}\} \leq \max\{\{QUAL_{P_j,s}(t') \mid j = 1, \ldots, d\}\}$
   $\Leftrightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$

and

2. for the pessimistic quality valuation:
   $\min\{\{QUAL_{P_j,s}(t) \mid j = 1, \ldots, d\}\} \leq \min\{\{QUAL_{P_j,s}(t') \mid j = 1, \ldots, d\}\}$
   $\Leftrightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$. □

Corollary 3.1

For a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$ the optimistic quality valuation delivers the valuation 'sufficient' for a tuple t if and only if for all j = 1, …, d: $QUAL_{P_j,s}(t)$ = 'sufficient'. *

Corollary 3.2

For a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$ the pessimistic quality valuation delivers the valuation 'perfect' for a tuple t if and only if for all j = 1, …, d: $QUAL_{P_j,s}(t)$ = 'perfect'. *

Besides these two valuations often a more moderate approach is appropriate in various situations. Therefore, two statistically more robust techniques are introduced, followed by a counter-example that violates the postulate of Definition 3.3.

Definition 3.25 Equidistant linguistic average valuation of a Pareto preference result

Assume a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, in a situation s where for all P_j, j = 1, …, d, it holds that P_j is no Pareto preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The equidistant linguistic average valuation for a result tuple t is defined as follows regarding the order of quality terms of Definition 3.8:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & y = 5 \\ \text{'very good'}, & y = 4 \\ \text{'good'}, & y = 3 \\ \text{'acceptable'}, & y = 2 \\ \text{'sufficient'}, & y = 1 \end{cases}$$

$$\text{where } y := \text{round}\left(\frac{1}{d} \sum_{j=1}^{d} g(QUAL_{P_j,s}(t)) \right)$$

$$\text{where } g(x) := \begin{cases} 5, & x = \text{'perfect'} \\ 4, & x = \text{'very good'} \\ 3, & x = \text{'good'} \\ 2, & x = \text{'acceptable'} \\ 1, & x = \text{'sufficient'} \end{cases}$$

$$\text{and round}(x) := \begin{cases} \lceil x \rceil & \text{if } x - \lfloor x \rfloor \geq 0.5 \\ \lfloor x \rfloor & \text{if } x - \lfloor x \rfloor < 0.5 \end{cases}$$

*

Example 3.16 Sample equidistant linguistic average quality valuation

Considering again the sample preference of Example 3.15 and the single qualities as shown in Table 3.6, the quality with the equidistant linguistic average valuation for a tuple t is computed as follows:

$$y := \text{round}\left(\frac{4+2+4+4+4+2+3}{7} \right) = \text{round}\left(\frac{23}{7} \right) = 3$$, thus $QUAL_{P,s}(t)$ = 'good'.

Table 3.6 illustrates the average valuation of these equally important preferences.

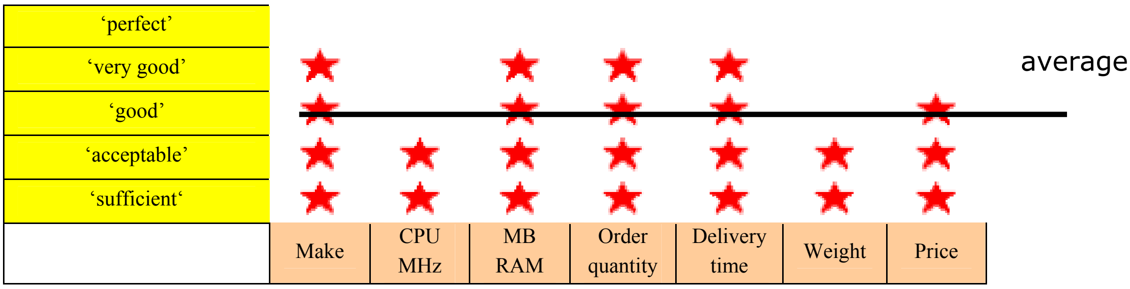


Table 3.6 Sample qualities of notebook characteristics for one result tuple t

⌂

Lemma 3.8

For the equidistant linguistic average quality valuation of a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, it holds that

$$\text{for all elements } t, t' \text{ with } t[A], t'[A] \in \mathrm{dom}(A)\text{: } t <_P t' \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t'). \qquad *$$

Proof 3.8

All preferences P_j satisfy the postulate of Definition 3.3. Thus, by $t <_P t'$ and the definition of a Pareto preference it holds that

$$\text{for all } j = 1, \ldots, d : QUAL_{P_j,s}(t) \leq QUAL_{P_j,s}(t').$$

Thus, because g(x) and round(x) are monotonically increasing the proposition holds. □

Definition 3.26 Median quality valuation of a Pareto preference result

Assume a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, in a situation s where for all P_j, $j = 1, \ldots, d$, it holds that P_j is no Pareto preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. Let $(QUAL_{P_1',s}(t), \ldots, QUAL_{P_d',s}(t))$ be the list with ascending ordered values of $QUAL_{P_j,s}(t)$. Then the median valuation for a result tuple t is defined as follows regarding to the order of quality terms of Definition 3.8:

$$QUAL_{P,s}(t) := \mathrm{median}(QUAL_{P_1',s}(t), \ldots, QUAL_{P_d',s}(t)),$$

where median(X) denotes the $\left\lfloor \frac{d+1}{2} \right\rfloor$th value of an ascending ordered list X consisting of d elements. *

Example 3.17 Sample median quality valuation for a Pareto preference

Using the same sample preference as in Example 3.15 and Table 3.5 the ordered list of quality statements would be (‘acceptable’, ‘acceptable’, ‘good’, ‘very good’, ‘very good’, ‘very good’, ‘very good’). The fourth value in this set is ‘very good’. This states the result of the median quality information function, which might be the appropriate valuation for Lisa’s situation. This function is noticeably more robust than the optimistic and pessimistic approaches mentioned before with respect to statistical outliers. The sample of Table 3.5 is given in ascending order in the following Table 3.7 for the visualization of this kind of valuation.

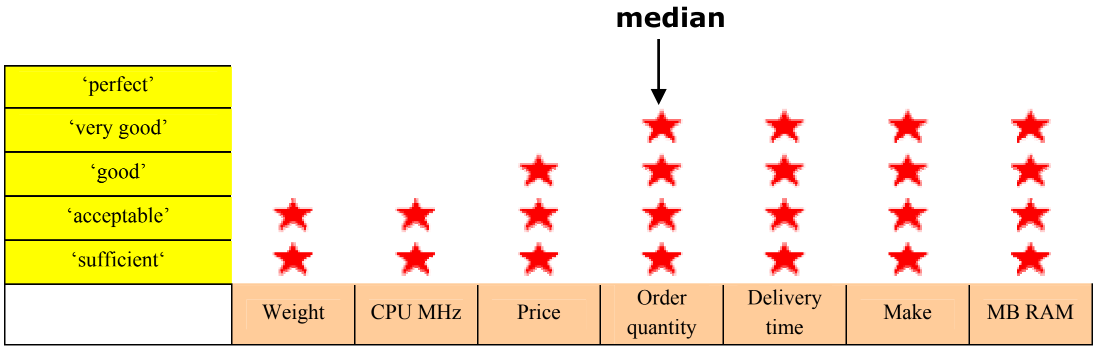


Table 3.7 Sample qualities of notebook characteristics in ascending order

⌂

Lemma 3.9

For the median quality valuation of a Pareto preference $P := P_1 \otimes \ldots \otimes P_d$, $d \geq 2$, it holds that

$$\text{for all elements } t, t' \text{ with } t[A], t'[A] \in dom(A): t <_P t' \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t'). \quad *$$

Proof 3.9

All preferences P_j satisfy the postulate of Definition 3.3. Thus, by $t <_P t'$ and the properties of a Pareto preference it holds that

$$\text{for all } j = 1, \ldots, d: QUAL_{P_j,s}(t) \leq QUAL_{P_j,s}(t')$$

$(QUAL_{P_{1'},s}(t), \ldots, QUAL_{P_{d'},s}(t))$ is the ascending ordered list of values $QUAL_{P_j,s}(t)$ and analogously $(QUAL_{P_{1''},s}(t'), \ldots, QUAL_{P_{d''},s}(t'))$.

$$\Rightarrow \text{for all } j' = j'', j', j'' \in \{1, \ldots, d\}: QUAL_{P_{j'},s}(t) \leq QUAL_{P_{j''},s}(t')$$
$$\Rightarrow median(QUAL_{P_{1'},s}(t), \ldots, QUAL_{P_{d'},s}(t)) \leq median(QUAL_{P_{1''},s}(t'), \ldots, QUAL_{P_{d''},s}(t'))$$
$$\Leftrightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t') \quad \Box$$

An -at first sight intuitive- approach is to use the statistical modus function for the quality valuation which is also robust with respect to statistical outliers. But it violates the postulate of Definition 3.3 as shown in the following counter-example and is therefore not appropriate for a believable and intuitive quality valuation.

Example 3.18 Counter-example: modus quality valuation for a Pareto preference

For a given Pareto preference $P := P_1 \otimes \ldots \otimes P_d$ and a result tuple t the modus valuation of a Pareto preference in a situation s could be defined as

$$QUAL_{P,s}(t) := modus(\{\{QUAL_{P_j,s}(t) \mid j = 1, \ldots, d\}\}),$$

where modus(X) is the most frequent value of the multi-set X. But considering once again the Pareto preference $P := P_{make} \otimes P_{mhz} \otimes P_{ram} \otimes P_{oquan} \otimes P_{del\ time} \otimes P_{weight} \otimes P_{price}$ of Example 3.15, the contradiction to an intuitive and comprehensible valuation becomes clear. In the following two tables the qualities of two given tuples t_1, t_2 with $t_1 <_P t_2$ are illustrated. Yet, the modus quality valuation would compute a 'very good' t_1 and only just a 'good' quality for the more preferred t_2, which obviously is a contradiction.

	Make	CPU MHz	MB RAM	Order quantity	Delivery time	Weight	Price
'perfect'							
'very good'					★	★	★
'good'			★	★	★	★	★
'acceptable'	★	★	★	★	★	★	★
'sufficient'	★	★	★	★	★	★	★

Table 3.8 Qualities of notebook characteristics for sample tuple t_1

	Make	CPU MHz	MB RAM	Order quantity	Delivery time	Weight	Price
'perfect'						★	★
'very good'					★	★	★
'good'		★	★	★	★	★	★
'acceptable'	★	★	★	★	★	★	★
'sufficient'	★	★	★	★	★	★	★

Table 3.9 Qualities of notebook characteristics for sample tuple t_2

⌂

Perhaps, one can design further intuitive valuations that fit further special situations. But this design must be done carefully with respect to the postulate of Definition 3.3.

3.4.2 Quality valuation of a prioritized preference

According to the philosophy of a prioritized preference $P_1 \& \ldots \& P_d$ the preference P_1 is the dominating preference and thus must have the decisive impact. Because $P_2, \ldots, P_d$ are subordinated. P_1 as the only decisive factor leads to the following valuation.

Definition 3.27 Quality valuation of a prioritized preference result

Assume a prioritized preference P := P_1 & … & P_d, d ≥ 2, in a situation s where for all P_j, j = 1, …, d, it holds that P_j is no prioritized preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The valuation for a result tuple t is defined as follows:

$$QUAL_{P,s}(t) := QUAL_{P_1,s}(t)$$ *

Example 3.19 Sample quality valuation for a prioritized preference

For Marge's next purchase due to her latest experiences she changed her mind and believes the most important factor for notebook reselling is the make. This is more important than the clock frequency, which itself is more important than the main memory. Then the subordinated attributes are the weight and last the price. In preference algebra this can be formulated as the prioritized preference

$$P := P_{make} \ \& \ P_{mhz} \ \& \ P_{ram} \ \& \ P_{weight} \ \& \ P_{price}.$$

In Table 3.10 the preference quality information for each P_j is given for one result tuple t. Analogue to Table 3.5, the stars denote the qualities for the preferences involved. With the quality valuation of Definition 3.27 the quality for P is computed simply as follows:

$$QUAL_{P,s}(t) := QUAL_{P_{make},s}(t) = \text{'good'}$$

dominating

	Make	CPU MHz	MB RAM	Weight	Price
'perfect'			★		★
'very good'		★	★	★	★
'good'	★	★	★	★	★
'acceptable'	★	★	★	★	★
'sufficient'	★	★	★	★	★

Table 3.10 Sample qualities referring to a prioritized preference

⌂

Lemma 3.10

For the quality valuation of a prioritized preference P := P_1 & … & P_d, d ≥ 2, according to Definition 3.27 it holds that

for all elements t, t' with t[A], t'[A] ∈ dom(A): $t <_P t' \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$. *

Proof 3.10

All preferences P_j satisfy the postulate of Definition 3.3. Thus, by $t <_P t'$ two cases can be considered for Definition 3.27.

1.) t' is preferred over t because of $P_2, \ldots, P_d$. Then
$QUAL_{P_1,s}(t) = QUAL_{P_1,s}(t') \Rightarrow QUAL_{P,s}(t) = QUAL_{P,s}(t')$
2.) t' is preferred over t because of P_1. Then
$QUAL_{P_1,s}(t) \leq QUAL_{P_1,s}(t') \Rightarrow QUAL_{P,s}(t) \leq QUAL_{P,s}(t')$ □

The valuation of Definition 3.27 is the only reasonable. A further correct valuation regarding the postulate of Definition 3.3 would be to valuate all result tuples t with the same quality. Although this just satisfies the postulate it is not comprehensible for a customer. Moreover this valuation would contradict the general model of the $QUAL_{P,s}$-function as defined in Definition 3.9, demanding that at least the first and the last partition are not empty. It would be possible but also not very intuitive to constantly shift the quality of a prioritized preference one level higher or lower than defined in Definition 3.27. The question still left is whether there is a correct valuation with a real impact, i.e. no constant valuation or no constant shift of the valuation, of one or more subordinated preferences $P_2, \ldots, P_d$. The following Lemma 3.11 shows that there is no further correct valuation.

Lemma 3.11

For the valuation of a prioritized preference $P := P_1 \,\&\, \ldots \,\&\, P_d$ there is no general correct valuation regarding the postulate of Definition 3.3 with a real impact of $QUAL_{P_2,s}, \ldots, QUAL_{P_d,s}$. *

Proof 3.11

This proof is performed for the case $P := P_1 \,\&\, P_2$. The generalization to $P_1 \,\&\, \ldots \,\&\, P_d$ is obvious. The first assumption is that there could be a correct quality valuation without impact of $QUAL_{P_1,s}(t)$. This means the valuation would be the following: Given is a prioritized preference $P := P_1 \,\&\, P_2$ in a situation s, where for all P_j, j = 1, 2 it holds that $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The valuation for a result element t in a situation s would be defined as:
$QUAL_{P,s}(t) := QUAL_{P_2,s}(t)$.
The following counter-example shows the contradiction to the postulate of Definition 3.3. Considering two possible elements t, t' with the following properties:
a) $t <_P t'$
b) $QUAL_{P_2,s}(t) > QUAL_{P_2,s}(t')$
Because of b) and the suggested valuation approach $\Rightarrow QUAL_{P,s}(t) > QUAL_{P,s}(t')$. Because of a) this is a contradiction according to the quality postulate of Definition 3.3.

Thus, $QUAL_{P_1,s}(t)$ must have an impact on $QUAL_{P,s}(t)$. It is to be shown that $QUAL_{P_1,s}(t)$ has indeed the only real impact. Assuming in case of $QUAL_{P_1,s}(t) < QUAL_{P_2,s}(t)$ that the quality of $QUAL_{P,s}(t)$ can be shifted to a higher quality level than $QUAL_{P_1,s}(t)$. The following counter-example shows the contradiction. Considering two possible elements t, t' with the following properties:
1.) $t <_{P_1} t'$
2.) $QUAL_{P_1,s}(t) < QUAL_{P_2,s}(t)$
3.) $QUAL_{P_1,s}(t') > QUAL_{P_2,s}(t')$
4.) $QUAL_{P_1,s}(t) = QUAL_{P_1,s}(t')$

With these properties and the suggestion to rise the quality in case of $QUAL_{P_1,s}(t) < QUAL_{P_2,s}(t)$ for all elements t the contradiction according to the quality postulate of Definition 3.3 will be shown: *Because of the partitioning* within the $QUAL_{P,s}$-function both have the same valuation regarding P_1 (4.)), even though t' is preferred over t according to P_1 (1.)). $QUAL_{P,s}(t')$ would be $QUAL_{P_1,s}(t')$, because of 3.). $QUAL_{P,s}(t)$ would be shifted to a higher quality than $QUAL_{P_1,s}(t)$, because of 2.). That means:

$$QUAL_{P,s}(t') = QUAL_{P_1,s}(t') = QUAL_{P_1,s}(t) < QUAL_{P,s}(t)$$

$$\Rightarrow QUAL_{P,s}(t') < QUAL_{P,s}(t)$$

That violates the quality postulate of Definition 3.3, because of 1.) $\Rightarrow t <_P t'$. t' must not be valuated worse than t $\Rightarrow$ contradiction.

Shifting the quality to a lower level for a lower $QUAL_{P_2,s}(t)$ fails analogously. For mathematical completeness also the following non-intuitive cases would fail analogously, i.e. to shift the overall quality higher than $QUAL_{P_1,s}(t)$ because of a lower or equal $QUAL_{P_2,s}(t)$ and vice versa. Thus, the impact of $QUAL_{P_2,s}(t)$ would violate the quality postulate in general.

Last point to clear is whether there is such a pair of tuples. The following brief examples for both intuitive cases bring the evidence. Assume properties of audio CDs, namely the price and the runtime, a given search preference $P := P_1 \ \& \ P_2$, with $P_1 :=$ LOWEST(price) and $P_2 :=$ HIGHEST(runtime). The $QUAL_{P,s}$-functions might be the following:

$$QUAL_{P_1,s}(t) := \begin{cases} \text{'perfect'} , & \text{price} \le 10 \\ \text{'very good'} , & 10 < \text{price} \le 14 \\ \text{'good'} , & 14 < \text{price} \le 16 \\ \text{'acceptable'}, & 16 < \text{price} \le 18 \\ \text{'sufficient'} , & 18 < \text{price} \end{cases}$$

$$QUAL_{P_2,s}(t) := \begin{cases} \text{'perfect'} , & 65 \le \text{runtime} \\ \text{'very good'} , & 55 < \text{runtime} \le 65 \\ \text{'good'} , & 45 < \text{runtime} \le 55 \\ \text{'acceptable'}, & 35 < \text{runtime} \le 45 \\ \text{'sufficient'} , & \text{runtime} < 35 \end{cases}$$

Obviously the two sample tuples t := (13, 70) and t' := (12, 45) satisfy the conditions of 1.) till 4.). If the quality of t would be shifted to a better quality because of the 'perfect' runtime then it would receive a better valuation than t' even though t' is preferred over t because of the better price.

Exactly the same example can be used to show the contradiction for the failing dual approach of shifting the quality to a lower level if the subordinated preferences are of lower quality. Tuple t' would be shifted to a lower quality than 'very good', whereas t would be valuated with 'very good'. □

Please note that the information of $QUAL_{P_2,s}(t)$, …, $QUAL_{P_d,s}(t)$ of a prioritized preference is not necessary for the quality of the prioritized preference, but *provides valuable arguments* for a result presentation.

3.4.3 Quality valuation of a numerical preference

As described in Definition 2.9 the numerical preference is an accumulation of SCORE preferences ranked by a combining function. Thus, the valuation is naturally given analogously to the one of the SCORE preference by separating the ranges of the combining function.

Definition 3.28 $QUAL_{P,s}$-function of a numerical preference

Assume a numerical preference P := $rank_F(P_1, P_2, \ldots, P_d)$ in a situation s, where for all P_j, j = 1, …, d it holds that P_j is no numerical preference and $QUAL_{P_j,s}(t)$ fulfills the postulate of Definition 3.3. The valuation for a result tuple t is defined as follows:

$$QUAL_{P,s}(t) := \begin{cases} \text{'perfect'}, & b_1(s) \le F(f_1(x_1), f_2(x_2), \ldots, f_d(x_d)) \\ \text{'very good'}, & b_2(s) \le F(f_1(x_1), f_2(x_2), \ldots, f_d(x_d)) < b_1(s) \\ \text{'good'}, & b_3(s) \le F(f_1(x_1), f_2(x_2), \ldots, f_d(x_d)) < b_2(s) \\ \text{'acceptable'}, & b_4(s) \le F(f_1(x_1), f_2(x_2), \ldots, f_d(x_d)) < b_3(s) \\ \text{'sufficient'}, & F(f_1(x_1), f_2(x_2), \ldots, f_d(x_d)) < b_4(s) \end{cases}$$

*

Example 3.20 Sample quality valuation for a numerical preference

Moe has success with his chart analysis (see Example 3.2). Moe's service is enhanced by two more key data. The exact definition is not of interest at this point. In Table 3.11 the results of the functions of the according SCORE preferences are shown. Also the results of the combining function

$$F(f_1(x, y, z), f_2(x, y, z), f_3(x, y, z)) := 2 * f_1(x, y, z) + f_2(x, y, z) + 3 * f_3(x, y, z)$$

for the numerical preference P are listed, by which means Moe suggests very valuable shares.

	enterprise	x	y	z	$f_1(x,y,z)$	$f_2(x,y,z)$	$f_3(x,y,z)$	F(…)
t_1	Coal & Quarry	5,07	39842.40	5108000	0.0395	5.4343	-0.1115	5.1788
t_2	Alaska Beer	-3,08	369783.18	27761500	-0.0410	2.2223	-0.3110	1.2073
t_3	Zodiac Inc.	12,23	868375.20	7459200	1.4238	3.0988	1.5687	10.6525
t_4	Nozzle Prod.	1,72	213999.12	1285280	0.2864	8.3322	1.2234	12.5752

Table 3.11 Sample excerpt of a stock exchange database

For the recommendation service to his favorite customer Homer Moe has designed the following $QUAL_{P,s}$-function for P:

$$\text{QUAL}_{P,s}(t) := \begin{cases} \text{'perfect'} , & 11 \le F(f_1(x,y,z), f_2(x,y,z), f_3(x,y,z)) \\ \text{'very good'} , & 8 \le F(f_1(x,y,z), f_2(x,y,z), f_3(x,y,z)) < 11 \\ \text{'good'} , & 4 \le F(f_1(x,y,z), f_2(x,y,z), f_3(x,y,z)) < 8 \\ \text{'acceptable'}, & 0 \le F(f_1(x,y,z), f_2(x,y,z), f_3(x,y,z)) < 4 \\ \text{'sufficient'} , & F(f_1(x,y,z), f_2(x,y,z), f_3(x,y,z)) < 0 \end{cases}$$

Today, Moe could recommend a perfect result within his recommendation to Homer via online service, namely Nozzle Prod. ⌂

Lemma 3.12

For the quality valuation of a numerical preference according to Definition 3.28 it holds that

for all elements t, t' with t[A], t'[A] $\in$ dom(A): $t <_P t' \Rightarrow \text{QUAL}_{P,s}(t) \le \text{QUAL}_{P,s}(t')$ *

Proof 3.12

This proof can be done analogously to Proof 3.1 a) of the SCORE preference. □

3.4.4 Quality valuation of grouped preferences

σ[P groupby A](R) := $\sigma[A^{\leftrightarrow}\&P](R)$ as specified in Definition 2.11 denotes the grouping of preferences, that means, partitioned for each value of dom(A) the preference P is evaluated over the underlying database relation. As described in section 3.3.14 there is no quality valuation of an ANTI-CHAIN preference. For the grouped preferences that means each result tuple t in a situation s is valuated with $\text{QUAL}_{P,s}(t)$.

3.4.5 Calculation of $\text{QUAL}_{P,s}$

In the sections 3.3 and 3.4 the quality valuation functions were defined for each base and complex preference in the form $\text{QUAL}_{P,s}(t)$. The following algorithm uses these definitions to calculate the qualities of any given Preference P. In this, the structure of the complex preferences is used to calculate recursively the quality of each combined preference. Thus, all presentation arguments regarding the quality are computed by this algorithm. With this algorithm the complex attribute Q of BMO^+ of Definition 3.4 can be calculated.

Algorithm 3.1 Calculation of the quality valuations of a search result

Assume a preference P and the $\text{QUAL}_{P,s}$-functions for all preferences, which are combined in P, for the situation s. Then the calculation of the quality valuations for a result tuple t $\in$ BMO is defined as follows:

0: calc($\text{QUAL}_{P,s}(t)$) :=
1: **case** P is a base preference
2: **then return** $\text{QUAL}_{P,s}(t)$ according to Definition 3.10 - Definition 3.20;

3: **case** P is a prioritized preference of the form $P := P_1 \& \dots \& P_d$
4: **then for** $j = 1, \dots, d$ **do** calc($QUAL_{P_j,s}(t)$) **and**
5: **return** $QUAL_{P,s}(t)$ according to Definition 3.27;
6: **case** P is a Pareto preference of the form $P := P_1 \otimes \dots \otimes P_d$
7: **then for** $j = 1, \dots, d$ **do** calc($QUAL_{P_j,s}(t)$) **and**
8: **return** $QUAL_{P,s}(t)$ according to Definition 3.23 - Definition 3.26;
9: **case** P is a numerical preference of the form $P := rank_F(P_1, P_2, \dots, P_d)$
10: **then for** $j = 1, \dots, d$ **do** calc($QUAL_{P_j,s}(t)$) **and**
11: **return** $QUAL_{P,s}(t)$ according to Definition 3.28; *

Theorem 3.1 Properties and correctness of Algorithm 3.1

For Algorithm 3.1 it holds that:

a) It computes the overall quality for a tuple $t \in BMO$ for the preference P in a situation s.
b) It computes the qualities of all preferences, which are involved in P, in a situation s for a tuple $t \in BMO$.
c) All involved base preference qualities hold the postulate of Definition 3.3.
d) All involved complex preferences hold the postulate of Definition 3.3.
e) The algorithm terminates correctly. *

Proof 3.13

The propositions of Theorem 3.1 are correct:

a) This is the return value of the algorithm, thus if b) holds, then also this is proved.
b) If P of the input parameter $QUAL_{P,s}(t)$ is a base preference, then in line 2 the correct value is returned. Otherwise P is complex and its quality is computed in one case of the lines 3, 6, or 9. There the qualities of all base preferences, which are accumulated on this level, are computed with one call of calc(..). Qualities of accumulated complex preferences are computed recursively.
c) This is immediately given by the exclusive usage of Definition 3.10 - Definition 3.20 and proved in Lemma 3.1 - Lemma 3.6.
d) This is immediately given by the exclusive usage of Definition 3.23 - Definition 3.26, Definition 3.27, and Definition 3.28 and proved in Lemma 3.7 - Lemma 3.12.
e) If P of the input parameter $QUAL_{P,s}(t)$ is a base preference, then it terminates. Otherwise the complex preference P terminates when the recursive call receives back all the sub-calls. They all terminate in the case of a base preference, because a complex preference is an accumulation of complex and base preferences and its structure is finite (Noetherian induction). □

Lemma 3.14

The complexity of Algorithm 3.1 is $\mathcal{O}(n)$, where n is the number of base preferences which are accumulated in the input preference P. *

Proof 3.14

Computing the quality for a tuple t according to a base preference constructor is supposed to be a constant effort, because only the affiliation to one of the five partitions must be determined. This must be done n times for n involved base preferences. The complexity for a numerical complex preference is analogously constant and for a prioritized preference the effort is only one step, the projection to the first preference involved. These valuations can be necessary at most n times. The effort for the quality calculation of a Pareto preference depends on the underlying valuation approach. Considering the most expensive case if all base preferences are accumulated within one preference, all introduced approaches must be analyzed. The pessimistic and optimistic valuation need simply a minimum, respective a maximum search which also leads to $\mathcal{O}(n)$. The equidistant approach needs n-1 additions and one dividing and thus $\mathcal{O}(n)$. The median can efficiently be computed in $\mathcal{O}(n)$ as shown in [AHU74]. Thus, independent of the choice of the introduced approaches the calculation can be done in $\mathcal{O}(n)$. □

Finally, for a complex nested preference a sample computation is given.

Example 3.21 Sample for a nested complex preference result

Marge once again wants to order new notebooks from Apu. She has learnt from her experience and has now a much better knowledge what could be sold easily. Thus, Marge has four preferences of first priority. These four **equally important** preferences are:

- Marge **likes** the notebook makes HP **and** Toshiba most. But she **avoids** Sony.
- About the main memory she **prefers** the **highest** possible capacity which is **more important** than her preference about the main memory make. There she **likes** Infineon **and** Samsung **but avoids** Kingston.
- About the CPU Marge **likes** Intel **more than** AMD which is **more important** than the clock frequency which **should be around** 2 GHz. And this is **more important** than the chip set, which **should be** a slot 2.
- And of course she **wants** to have a very **low** notebook price.

These preferences are **more important** for Marge than the following three **equally important** preferences of second priority:

- She **prefers** the **lowest** weight for a notebook.
- Moreover, she **prefers** a warranty period **around** 3 years which is **more important** than an included home repair service.
- About the delivery terms Marge **prefers** a very **fast** delivery which is **more important** than the order quantity, which **should be around** 40.

The formal sample construction of the described preference(s) based on the preference constructors of [Kie02, Kie04] could be expressed as follows:

- P_{nbm} := POS/NEG(notebook make, {HP, Toshiba}; {Sony})
- P_{mmc} := HIGHEST(main memory capacity)

- P_{mmm} := POS/NEG(main memory make, {Infineon, Samsung}; {Kingston})
- P_{mm} := P_{mmc} & P_{mmm}
- P_{cm} := POS/POS(CPU make, {Intel}; {AMD})
- P_{ccf} := AROUND(clock frequency, 2)
- P_{ccs} := POS(chip set, {slot 2})
- P_{cpu} := P_{cm} & P_{ccf} & P_{ccs}
- P_{pr} := LOWEST (price)
- $P_{priority1}$:= $P_{nbm} \otimes P_{mm} \otimes P_{cpu} \otimes P_{pr}$
- P_{we} := LOWEST (weight)
- P_{wap} := AROUND(warranty period, 3)
- P_{hrs} := POS(home repair service, {yes})
- P_{wa} := P_{wap} & P_{hrs}
- P_{dti} := LOWEST(delivery time)
- P_{oq} := AROUND(order quantity, 40)
- P_{dte} := P_{dti} & P_{oq}
- $P_{priority2}$:= $P_{we} \otimes P_{wa} \otimes P_{dte}$
- P := $P_{priority1}$ & $P_{priority2}$

Only using the just constructed base preferences P is engineered as

$$P := (P_{nbm} \otimes (P_{mmc} \,\&\, P_{mmm}) \otimes (P_{cm} \,\&\, P_{ccf} \,\&\, P_{ccs}) \otimes P_{pr}) \,\&\, (P_{we} \otimes (P_{wap} \,\&\, P_{hrs}) \otimes (P_{dti} \,\&\, P_{oq})).$$

This preference construction is schematically illustrated in Figure 3.4 with the according domains, where each single box represents a base preference and each surrounding box stands for a complex preference. There, boxes side by side visualize a Pareto preference. A box illustrated immediately on top of another is one level higher within a prioritized preference.

Apu wants to enable his e-procurement platform for his resellers to do good reasoning during the sales dialog. Therefore, for the computation of the preference quality in Marge's situation at first the $QUAL_{P,s}$-functions for each base preference must be defined. Please note that even when the quality of a base preference has no impact on the overall quality it is nevertheless necessary to know about this quality, because it contains an argument for the result presentation. In the sequel, for a better understanding relative or parametric designs of the $QUAL_{P,s}$-functions are already filled with absolute data.

<u>P_{nbm} := POS/NEG(notebook make, {HP, Toshiba}; {Sony}):</u>

$$QUAL_{P_{nbm},s}(t) := \begin{cases} \text{'perfect'} & , \text{ notebook make} \in \{\text{Toshiba, HP}\} \\ \text{'good'} & , \text{ notebook make} \notin \{\text{Toshiba, HP, Sony}\} \\ \text{'sufficient'}, & \text{ notebook make} \in \{\text{Sony}\} \end{cases}$$

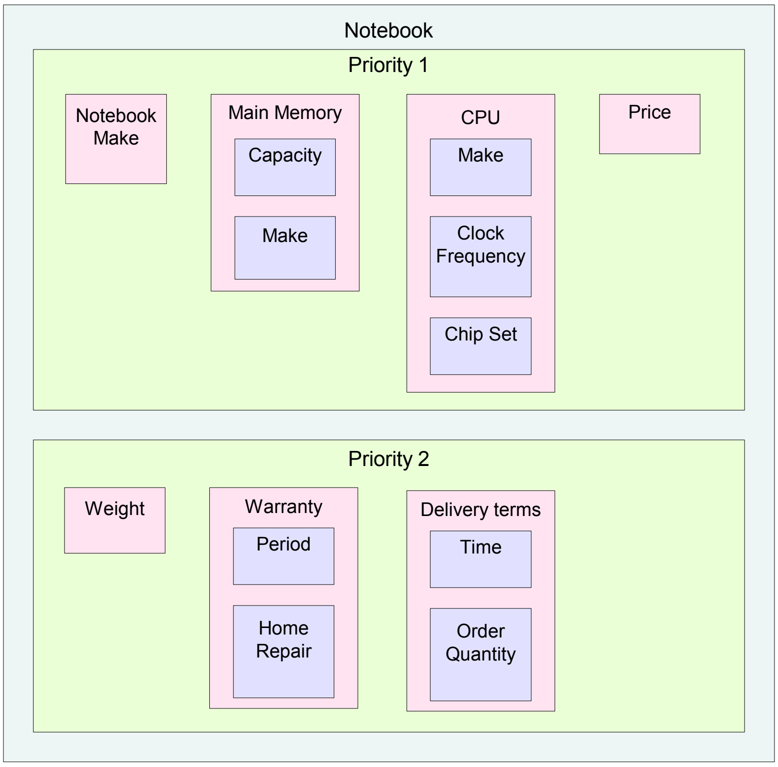


Figure 3.4 A nested complex preference example

For this and the following POS/NEG preferences Apu has decided to apply the moderate approach for the $QUAL_{P,s}$-function, i.e. elements which are neither in the POS-set nor in the NEG-set are valuated as 'good'. For the design of the $QUAL_{P,s}$-functions for the following numerical base preferences Apu uses his knowledge about the typical sensibilities of female resellers like Marge.

P_{mmc} := HIGHEST(main memory capacity):

$$QUAL_{P_{mmc},s}(t) := \begin{cases} \text{'perfect'}, & 1024 \le \text{main memory capacity} \\ \text{'very good'}, & 768 \le \text{main memory capacity} < 1024 \\ \text{'good'}, & 512 \le \text{main memory capacity} < 768 \\ \text{'acceptable'}, & 256 \le \text{main memory capacity} < 512 \\ \text{'sufficient'}, & \text{main memory capacity} < 256 \end{cases}$$

P_{mmm} := POS/NEG(main memory make, {Infineon, Samsung}; {Kingston}):

$$QUAL_{P_{mmm},s}(t) := \begin{cases} \text{'perfect'}, & \text{main memory make} \in \{\text{Infineon, Samsung}\} \\ \text{'good'}, & \text{main memory make} \notin \{\text{Infineon, Samsung, Kingston}\} \\ \text{'sufficient'}, & \text{main memory make} \in \{\text{Kingston}\} \end{cases}$$

Concerning the CPU make, Apu has the experience that all his customers normally are also very satisfied with a product of the alternative set of the POS/POS preference. Therefore he has applied the optimistic approach and designed the following $QUAL_{P,s}$-function:

P_{cm} := POS/POS(CPU make, {Intel}; {AMD}):

$$QUAL_{P_{cm},s}(t) := \begin{cases} \text{'perfect'}, & \text{CPU make} \in \{\text{Intel}\} \\ \text{'very good'}, & \text{CPU make} \in \{\text{AMD}\} \\ \text{'sufficient'}, & \text{CPU make} \notin \{\text{Intel, AMD}\} \end{cases}$$

P_{ccf} := AROUND(clock frequency, 2):

$$QUAL_{P_{ccf},s}(t) := \begin{cases} \text{'perfect'}, & \text{clock frequency} = 2.0 \\ \text{'very good'}, & 1.8 \le \text{clock frequency} < 2.0 \vee 2.0 < \text{clock frequency} \le 2.2 \\ \text{'good'}, & 1.6 \le \text{clock frequency} < 1.8 \vee 2.2 < \text{clock frequency} \le 2.4 \\ \text{'acceptable'}, & 1.4 \le \text{clock frequency} < 1.6 \vee 2.4 < \text{clock frequency} \le 2.6 \\ \text{'sufficient'}, & \text{clock frequency} < 1.4 \vee 2.6 < \text{clock frequency} \end{cases}$$

P_{ccs} := POS(chip set, {slot 2}):

$$QUAL_{P_{ccs},s}(t) := \begin{cases} \text{'perfect'}, & \text{chip set} \in \{\text{slot 2}\} \\ \text{'sufficient'}, & \text{chip set} \notin \{\text{slot 2}\} \end{cases}$$

P_{pr} := LOWEST(price):

$$QUAL_{P_{pr},s}(t) := \begin{cases} \text{'perfect'}, & \text{price} \le 1600 \\ \text{'very good'}, & 1600 < \text{price} \le 2000 \\ \text{'good'}, & 2000 < \text{price} \le 2800 \\ \text{'acceptable'}, & 2800 < \text{price} \le 3600 \\ \text{'sufficient'}, & 3600 < \text{price} \end{cases}$$

P_{we} := LOWEST(weight):

$$QUAL_{P_{we},s}(t) := \begin{cases} \text{'perfect'}, & \text{weight} \le 1.1 \\ \text{'very good'}, & 1.1 < \text{weight} \le 2.2 \\ \text{'good'}, & 2.2 < \text{weight} \le 2.9 \\ \text{'acceptable'}, & 2.9 < \text{weight} \le 3.5 \\ \text{'sufficient'}, & 3.5 < \text{weight} \end{cases}$$

P_{hrs} := POS(home repair service, {yes}):

$$QUAL_{P_{hrs},s}(t) := \begin{cases} \text{'perfect'}, & \text{home repair service} \in \{\text{yes}\} \\ \text{'sufficient'}, & \text{home repair service} \notin \{\text{yes}\} \end{cases}$$

P_{wap} := AROUND(warranty period, 3):

$$QUAL_{P_{wap},s}(t) := \begin{cases} \text{'perfect'}, & \text{warranty period} = 3.0 \\ \text{'very good'}, & 2.5 \le \text{warranty period} < 3.0 \vee 3.0 < \text{warranty period} \le 3.5 \\ \text{'good'}, & 2.0 \le \text{warranty period} < 2.5 \vee 3.5 < \text{warranty period} \le 4.0 \\ \text{'acceptable'}, & 1.0 \le \text{warranty period} < 2.0 \vee 4.0 < \text{warranty period} \le 5.0 \\ \text{'sufficient'}, & \text{warranty period} < 1.0 \vee 5.0 < \text{warranty period} \end{cases}$$

P_{dti} := LOWEST(delivery time):

$$QUAL_{P_{dti},s}(t) := \begin{cases} \text{'perfect'}, & \text{delivery time} \le 3 \\ \text{'very good'}, & 3 < \text{delivery time} \le 5 \\ \text{'good'}, & 5 < \text{delivery time} \le 9 \\ \text{'acceptable'}, & 9 < \text{delivery time} \le 13 \\ \text{'sufficient'}, & 13 < \text{delivery time} \end{cases}$$

P_{oq} := AROUND(order quantity, 40):

$$QUAL_{P_{oq},s}(t) := \begin{cases} \text{'perfect'}, & \text{order quantity} = 40 \\ \text{'very good'}, & 35 \le \text{order quantity} < 40 \vee 40 < \text{order quantity} \le 45 \\ \text{'good'}, & 30 \le \text{order quantity} < 35 \vee 45 < \text{order quantity} \le 50 \\ \text{'acceptable'}, & 20 \le \text{order quantity} < 30 \vee 50 < \text{order quantity} \le 60 \\ \text{'sufficient'}, & \text{order quantity} < 20 \vee 60 < \text{order quantity} \end{cases}$$

The application of Marge's preferences with the selection σ[P](R) using Apu's notebook database relation R delivers the BMO results t_i shown in Table 3.12.

nbm	Toshiba	HP	Toshiba	Dell	Dell	Sony
mmc	512	128	512	1024	1024	1024
mmm	Infineon	Kingston	Samsung	Kingston	Micron	Micron
cm	Intel	AMD	AMD	VIA	VIA	AMD
ccf	2.4	2.0	2.0	1.8	1.4	1.8
ccs	Slot 2	Slot 2	Slot 2	Socket 7	Slot 2	Socket 7
pr	3649	2333	3011	1699	1711	1599
we	3.0	2.0	1.9	1.0	1.5	3.1
wap	3	3	3	2.5	3	3
hrs	yes	yes	yes	no	no	yes
dti	7	8	12	3	4	5
oq	45	35	60	45	23	42
	t_1	t_2	t_3	t_4	t_5	t_6

Table 3.12 Sample BMO result set for a complex nested preference

Considering e.g. result tuple t_1, the qualities computed with the above mentioned $QUAL_{P,s}$-functions for each base preference lead to

- $QUAL_{P_{nbm},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{mmc},s}(t_1)$ = ‘good’
- $QUAL_{P_{mmm},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{cm},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{ccf},s}(t_1)$ = ‘good’
- $QUAL_{P_{ccs},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{pr},s}(t_1)$ = ‘sufficient‘
- $QUAL_{P_{we},s}(t_1)$ = ‘acceptable’
- $QUAL_{P_{wap},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{hrs},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{dti},s}(t_1)$ = ‘good’
- $QUAL_{P_{oq},s}(t_1)$ = ‘very good’

which is illustrated in Figure 3.5.

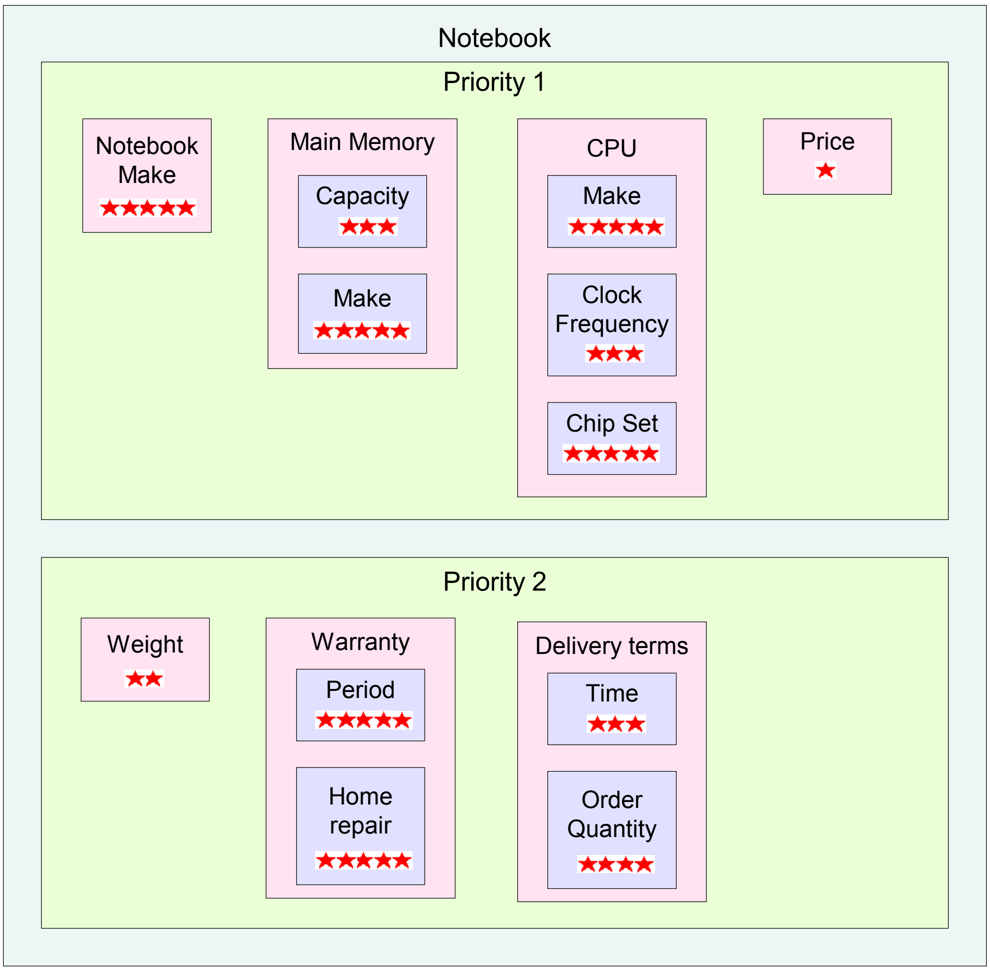


Figure 3.5 Quality valuations of base preferences for result t_1

Apu knows about Marge’s objective opinions. Therefore he valuates Pareto preferences with the median quality valuation. Thus, the quality for complex preferences in this example is computed recursively as follows:

- $QUAL_{P_{mm},s}(t_1)$ = ‘good’
- $QUAL_{P_{cpu},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{wa},s}(t_1)$ = ‘perfect’
- $QUAL_{P_{dte},s}(t_1)$ = ‘good’

- $QUAL_{P_{priority1},s}(t_1)$ = ‘good’
- $QUAL_{P_{priority2},s}(t_1)$ = ‘good’

- $QUAL_{P,s}(t_1)$ = ‘good’

Thus, as an overall impression, the result tuple t_1 is valuated as ‘good’. As declared in Definition 3.4, BMO is extended with the quality information by a complex attribute Q to BMO$^+$. To complete the picture and as a sample basis for the next two sections in Table 3.13 the single values of Q are computed for the sample BMO of Table 3.12.

P_{nbm}	‘perfect‘	‘perfect‘	‘perfect‘	‘good‘	‘good‘	‘sufficient‘
P_{mmc}	‘good‘	‘sufficient‘	‘good‘	‘perfect‘	‘perfect‘	‘perfect‘
P_{mmm}	‘perfect‘	‘sufficient‘	‘perfect‘	‘sufficient‘	‘good‘	‘good‘
P_{cm}	‘perfect‘	‘very good‘	‘very good‘	‘sufficient‘	‘sufficient‘	‘very good‘
P_{ccf}	‘good‘	‘perfect‘	‘perfect‘	‘very good‘	‘acceptable‘	‘very good‘
P_{ccs}	‘perfect‘	‘perfect‘	‘perfect‘	‘sufficient‘	‘perfect‘	‘sufficient‘
P_{pr}	‘sufficient‘	‘good‘	‘acceptable‘	‘very good‘	‘very good‘	‘perfect‘
P_{we}	‘acceptable‘	‘very good‘	‘very good‘	‘perfect‘	‘very good‘	‘acceptable‘
P_{wap}	‘perfect‘	‘perfect‘	‘perfect‘	‘very good‘	‘perfect‘	‘perfect‘
P_{hrs}	‘perfect‘	‘perfect‘	‘perfect‘	‘sufficient‘	‘sufficient‘	‘perfect‘
P_{dti}	‘good‘	‘good‘	‘acceptable‘	‘perfect‘	‘very good‘	‘very good‘
P_{oq}	‘very good‘	‘very good‘	‘acceptable‘	‘very good‘	‘acceptable‘	‘very good‘
P_{mm}	‘good‘	‘sufficient‘	‘good‘	‘perfect‘	‘perfect‘	‘perfect‘
P_{cpu}	‘perfect‘	‘very good‘	‘very good‘	‘sufficient‘	‘sufficient‘	‘very good‘
P_{wa}	‘perfect‘	‘perfect‘	‘perfect‘	‘very good‘	‘perfect‘	‘perfect‘
P_{dte}	‘good‘	‘good‘	‘acceptable‘	‘perfect‘	‘very good‘	‘very good‘
$P_{priority1}$	‘good‘	‘good‘	‘good‘	‘good‘	‘good‘	‘very good‘
$P_{priority2}$	‘good‘	‘very good‘	‘very good‘	‘perfect‘	‘very good‘	‘very good‘
P	‘good‘	‘good‘	‘good‘	‘good‘	‘good‘	‘very good‘
	t_1	t_2	t_3	t_4	t_5	t_6

Table 3.13 The single quality values of Q of a sample BMO$^+$ for a complex nested preference

⌂

3.5 Filter criterion "but-only"

The quality valuations with respect to the base preferences as well as for the complex preferences have been constructed. This provides important information for the decision which results to present. At this point there must be a decision which results are visible for the customer and which results should better be hidden and never presented or presented in a second phase. There are various reasons for filter criteria BOF ("but-only" filter) to compute BMO* := $\sigma_{BOF}(BMO^{++})$ (see Definition 3.5). Three obvious criteria are introduced in the following Definition 3.29, but of course the framework is not limited to these.

Definition 3.29 Filter criteria for a "but-only" filter BOF

Three important factors for a filter over the search results are

- hidden preferences of the knowledge engineer,
- issues of presentation style,
- quality claims of a customer. *

E.g. in a web shop several vendor preferences could be integrated immediately in a search preference P. But there are also lots of criteria that can only be handled after the soft selection. The vendor wants to present only one result per manufacturer, but with highest possible quality for P.

In style guides ([Wie03]) how to present search results one recommendation always is not to present too many results at once. That means, if the quantity bound is declared as 'k' and there are more than *k* tuples in BMO^{++} a selection on this quantity must be done. Because the results are all best matches, *k* random picks are one appropriate criterion. By the way, this is similar to a 'top-k' ([BCG02]) search. The compatibility of a preference search based on the model of [Kie02, Kie04] to a 'top-k' search is shown in [LK02].

When presenting results also the quality claims personalized for each customer must be respected. Some customers are displeased if several of their search preferences could not be respected. This kind of BOF can be defined as follows.

Definition 3.30 "But-only" filter criterion for quality claims with respect to base preferences

$aq_{\text{'sufficient'}}$ denotes the frequency of the quality valuation 'sufficient' for a result tuple t regarding to any base preference constructor involved as defined in Definition 3.4. Respecting the quality claims of a customer one filter criterion for BMO^{++} can be defined as

$$aq_{\text{'sufficient'}} \leq x,$$

to avoid too many 'sufficient' valuations for base preferences, namely more than x. *

Example 3.22 Respecting quality claims of a customer regarding unfulfilled base preferences

When Marge is visiting Apu's e-procurement portal she is annoyed when receiving search results with more than three 'sufficient' quality valuations according to her base preferences. Considering the results of Table 3.13 the bound beyond which results to hide is given in Figure 3.6. In this example all results but t_4 would be presented, because t_4 has four 'sufficient' valuations, all other results have even less than three.

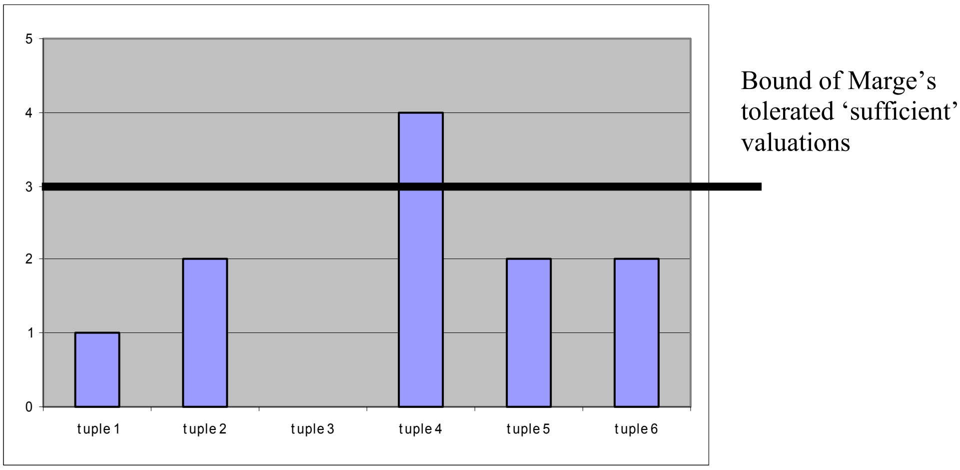


Figure 3.6 Number of 'sufficient' valuations of a sample BMO^{++}

⌂

This was a criterion with an absolute bound. Of course, also a relative threshold is imaginable. The next criterion considers the overall quality.

Definition 3.31 Respecting overall quality claims of a customer wrt. a search preference P

The overall quality of a result tuple t within BMO^{++} for the linguistic model of Definition 3.8 is instantiated as follows according to Definition 3.4:

$$aq_{overall} := \begin{cases} 5 & \text{if overall quality is 'perfect'} \\ 4 & \text{if overall quality is 'very good'} \\ 3 & \text{if overall quality is 'good'} \\ 2 & \text{if overall quality is 'acceptable'} \\ 1 & \text{if overall quality is 'sufficient'} \end{cases}$$

Respecting the overall quality claims of a customer a filter for BMO^{++} can be defined as

$$aq_{overall} \geq y,$$

where $y \in 1, \ldots, 5$. *

This instantiation of $aq_{overall}$ in BMO^{++} is also used in the sequel.

Example 3.23 Respecting quality claims of a customer wrt. a search preference P
Once again considering Table 3.13 and a customer's quality claim of at least 'acceptable' results, that means

$$aq_{overall} \geq 2,$$

all six tuples would satisfy this selection criterion, because they are all of at least 'good' quality. ⌂

Also combined approaches are imaginable, e.g. the knowledge engineer applies the "but-only" filter $aq_{overall} \geq 2$ if the price per unit is lower than 1000 €. Otherwise he takes the risk to annoy the customer in order to possibly realize a very high turnover. These two filter criteria have shown a personalized way how not to displease a customer. This aspect of sales psychology leads to the next section, the pointing out of one particular search result.

3.6 Selection criterion for pointing out a search result

Suppose the decision has been made which results to present to the customer. Still in many scenarios it is necessary *to point out one or more results*. The question which results are the best ones to present first is very situation and domain dependent. Agrawal and Wimmer ([AW00]) suggest bringing the results into a total order by rating the results with a sum over weighted preferences. This kind of presentation conforms to the common suggestion of Wiedemann ([Wie03]) to put the best results up front. But there is only a limited expressiveness in this search technology/preference model of Agrawal and Wimmer and also limited semantics for a smart presentation order. As shown in [Kie04, Cho03], the Pareto preference constructor and the prioritized preference constructor are not sub-constructors of the complex numerical preference constructor. I.e. with a ranking model like introduced in [AW00], a Pareto or prioritized preference cannot be formulated in general. Yet, this is necessary for several of the following presentation strategies. For pointing out a search result general rules for selection criteria are discussed, as well as some very domain dependant ones from the field of sales scenarios in e-procurement. It becomes clear that the preference model of [Kie02, Kie04] and the introduced quality valuation of search results form a very powerful framework for an easy and intuitive declaration of presentation preferences. Beside the *extensive expressiveness of this preference model* and *the ability to handle conflicts* a further important advantage, when using preferences instead of hard selection criteria, is that presentation strategies formulated as preferences *never produce the empty result effect*. That means, as an outcome of the presentation preference there will always be *at least one predestined result to point out*.

3.6.1 General selection criteria

One straight forward approach is to pick out a result with the best available overall quality, namely the highest $AQ_{overall}$.

Definition 3.32 Presentation preferences regarding the overall quality

Assume a result set BMO*. The presentation preference regarding the highest overall quality within BMO* is defined as:

$$PP := \text{HIGHEST}(AQ_{overall}) \qquad *$$

Example 3.24 Presentation preference with the highest overall quality

Considering the sample results of Table 3.13 the following overall qualities are given:

'perfect'						
'very good'						★
'good'	★	★	★	★	★	★
'acceptable'	★	★	★	★	★	★
'sufficient'	★	★	★	★	★	★
	t_1	t_2	t_3	t_4	t_5	t_6

Table 3.14 Sample of overall qualities of a BMO result set

Assuming the preference HIGHEST($AQ_{overall}$) with the most psychological advantage, t_6 would be the article to point out. ⌂

If this presentation preference PP delivers more than one result then a random pick out of these is an appropriate way. To get many presentation arguments for a good reasoning, it is very useful to pick out the results with the highest number of high quality single characteristics, i.e. with the most perfectly fulfilled base preferences. This leads to the next selection criterion.

Definition 3.33 Selection criterion with highest quantity of most positive result characteristics

Assume a result set BMO* with the aggregated attributes $AQ_{\text{'perfect'}}$, $AQ_{\text{'very good'}}$, $AQ_{\text{'good'}}$, and $AQ_{\text{'acceptable'}}$, counting the quality valuations with respect to the base preferences for a result t. Then the selection criterion for the highest quantity of most positive result characteristics is defined as the following complex presentation preference:

$$\begin{aligned} PP := {} & \text{HIGHEST}(AQ_{\text{'perfect'}}) \;\& \\ & \text{HIGHEST}(AQ_{\text{'very good'}}) \;\& \\ & \text{HIGHEST}(AQ_{\text{'good'}}) \;\& \\ & \text{HIGHEST}(AQ_{\text{'acceptable'}}) \end{aligned} \qquad *$$

The selection criterion of Definition 3.33 first picks out results with most 'perfectly' satisfied base preferences. If there is an equal number of 'perfect' statements then within these results the results with the most 'very good' valuations are delivered and so forth. Please note that base preferences that are subordinated in a prioritized preference have no impact on the overall quality. But indeed they are important presentation arguments and they have a real impact on this selection criterion.

Example 3.25 Sample selection for selection criterion of Definition 3.33

Referring to Table 3.13 the frequencies of the five possible quality terms are illustrated in Figure 3.7. With the presentation preference of Definition 3.33 the results t_1 and t_3 are best matches with respect to the accumulated preference of first priority. Because result t_1 has one 'very good' valuation less, t_3 is the one to point out. ⌂

The dual approach consists in avoiding negative impressions, which often displeases a customer. This approach is defined analogously in the sequel.

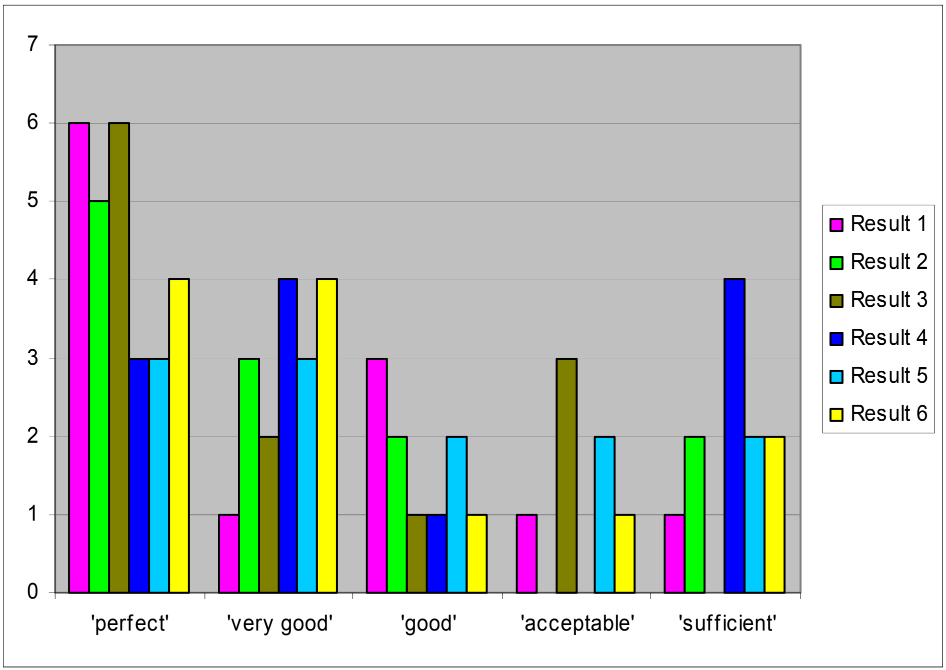


Figure 3.7 Sample frequencies of quality terms

Definition 3.34 Selection criterion with lowest quantity of most negative result characteristics

Assume a result set BMO* with the aggregated attributes $AQ_{\text{'very good'}}$, $AQ_{\text{'good'}}$, $AQ_{\text{'acceptable'}}$, and $AQ_{\text{'sufficient'}}$, counting the quality valuations regarding the base preferences for a result t. Then the selection criterion for the lowest quantity of most negative result characteristics is defined as the following complex presentation preference:

$$\begin{aligned} PP := {} & \text{LOWEST}(AQ_{\text{'sufficient'}}) \ \& \\ & \text{LOWEST}(AQ_{\text{'acceptable'}}) \ \& \\ & \text{LOWEST}(AQ_{\text{'good'}}) \ \& \\ & \text{LOWEST}(AQ_{\text{'very good'}}) \end{aligned}$$

*

Example 3.26 Sample selection for selection criterion of Definition 3.34

Considering the sample illustrated in Figure 3.7 result 3 is the only one with no 'sufficient' valuation and would therefore be selected according to the selection criterion of Definition 3.34. ⌂

Of course, also a Pareto combination of the last two mentioned dual presentation preference definitions is helpful to receive a more balanced impression. The last two presentation preferences have considered single characteristics. For some customers only the qualitative arguments are important. Therefore, a variant would be to consider only arguments of the first priority of a prioritized preference. The definition of the corresponding presentation preference is obvious. Another way to exploit the semantics of a prioritized preference $P := P_1 \ \& \ P_2$ is to prefer the overall quality of P_1, which is the overall quality of P, over the overall quality of P_2. That means, if there are several results with the same overall quality, then the results with the best quality regarding the second priority of the preference are preferred.

Definition 3.35 Selection criterion for a prioritized preference wrt. the priority level qualities

Assume a result set BMO* as the result of a search preference $P := P_1 \ \& \ P_2$ including $AQ_{overall\_P_1}$ and $AQ_{overall\_P_2}$, which are analogously defined to $AQ_{overall}$. The presentation preference PP regarding a preferred highest overall quality of the first priority level of P over the highest overall quality of the second priority level within BMO* is defined as:

$$PP := HIGHEST(AQ_{overall\_P_1}) \ \& \ HIGHEST(AQ_{overall\_P_2})$$ *

Example 3.27 Sample selection with the criterion of Definition 3.35

Considering the sample of Table 3.13 and the underlying search preference $P := P_{priority1} \ \& \ P_{priority2}$ all but result 6 have an overall valuation of ‘good’ quality for priority level one, result 6 is of ‘very good’ quality regarding priority one. Thus, result tuple 6 would be the one to point out.

Considering the same result set without result 6. Because all five results have a ‘good’ valuation regarding priority one, the best overall quality of the second priority is decisive. Result tuple 4 is the best one with a ‘perfect’ valuation regarding priority two. ⌂

A further possibility for a credible approach is to present a result where the single arguments confirm the overall impression.

Definition 3.36 Selection criterion with highest quantity of single valuations equal to overall valuation

Assume a result set BMO*. Let v_j denote the linguistic term of the overall quality. As defined in Definition 3.4 AQ_{v_j} denotes the frequency of base preferences, which are valuated with v_j. Then the selection criterion for the highest quantity of result characteristics equal to the overall quality is defined as the following presentation preference:

$$PP := HIGHEST(AQ_{v_j})$$

Please note that this must be considered separately for each occurring overall quality. *

Example 3.28 Sample selection with the criterion of Definition 3.36

Considering the sample of Table 3.13 all but result 6 have an overall valuation of 'good' quality, result 6 is of 'very good' quality. Thus, decisive for the results 1 - 5 is the corresponding number of 'good' single characteristics, for result 6 the number of 'very good' single characteristics. Figure 3.8 shows the corresponding number for each result. Result 6 delivers four confirming arguments and therefore is the best match of this presentation preference.

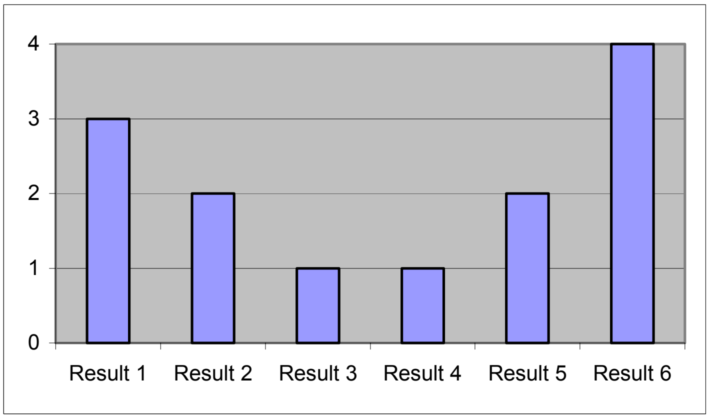


Figure 3.8 Quantity of single characteristics with equal quality to the overall quality

⌂

The above mentioned presentation preferences are intuitive standard rules ([Wie03]) that of course can be combined in any way. To prove that also the world of the weighted sum models, e.g. [AW00], can be satisfied with this framework the next approach is based on the SCORE preference and on the single qualities of the base preferences involved.

Definition 3.37 Selection criterion with combined ranked quality valuations

Assume a result set BMO* with the aggregated attributes $AQ_{\text{'perfect'}}$, $AQ_{\text{'very good'}}$, $AQ_{\text{'good'}}$, $AQ_{\text{'acceptable'}}$, and $AQ_{\text{'sufficient'}}$. Then the selection criterion with combined ranked quality valuations with respect to base preferences can be defined with the scoring function

$$f(t) := w_1 \cdot aq_{\text{'perfect'}} + w_2 \cdot aq_{\text{'very good'}} + w_3 \cdot aq_{\text{'good'}} + w_4 \cdot aq_{\text{'acceptable'}} + w_5 \cdot aq_{\text{'sufficient'}}$$

as the preference PP := SCORE(A, f) over the relation BMO*. *

This selection criterion is given to complete the picture for those knowledge engineers who work with the philosophy of weighted sums. All available results are best matches and therefore there might be a reason for any kind of scoring definition for the selection out of the best matches. One constructed example demonstrates a use case.

Example 3.29 Selection by combined ranked quality valuations

Considering the results of Table 3.13 and assuming an interest in a high quantity of fulfilled base preferences Definition 3.37 could be used as follows. Because of the interest in a high quantity of positive characteristics a knowledge engineer could define the weights as $w_1 = 10$, $w_2 = 9$, $w_3 = 7$, $w_4 = 2$, and $w_5 = 0$. For 'Result 1' the ranking function is calculated as follows:

$$f(t_1) = 10 \cdot 6 + 9 \cdot 1 + 7 \cdot 3 + 2 \cdot 1 + 0 \cdot 1 = 92.$$

In brief the other results, i.e. $f(t_2) = 91$, $f(t_3) = 91$, $f(t_4) = 73$, $f(t_5) = 75$, and $f(t_6) = 85$. With the highest score of 92 the result number 1 would be selected first in this scenario. ⌂

3.6.2 Presentation criteria in sales scenarios

As a special domain, selection criteria for a presentation within a sales scenario, e.g. an e-procurement store, are discussed in this section. Especially in a sales dialog it is very important to choose the adequate products to point out, i.e. to follow a special sales strategy ([Rac89, HS98]). The style of a good vendor is to actively recommend one or more results and to argue why this is an appropriate product. Thereby, the vendor can improve his believability when he is able to adequately valuate the alternatives ([Bec98]). And of course, the better the quality of a search result the higher is the psychological advantage for the vendor within a result presentation.

There is not the only perfect strategy to convince a customer. Many cognitive factors and factors of the environment have an impact on the success of a sales scenario, as is mentioned in most sales models and models of consumer choice behavior ([Bän85]). Some people can be convinced by a more blatant offering. Other people positively react to an offer, when positive and negative aspects of the search result are discussed. Also vendors have different utility functions. In this section, major aspects for the selection of the search result, which should be pointed out, are discussed. There can even be conflicting aspects, e.g. when there is a result in the result set with lots of high-quality presentation arguments and there is a further result with a higher profit for the vendor. Therefore, for each criterion it is shown how to easily select the adequate result. Each selection criterion is declaratively defined as a preference according to the model of [Kie02, Kie04]. As shown these preferences can flexibly be combined into more complex preferences, even if there are conflicts in the preferences. With the following definitions of criteria a flexible framework is given to support well-known aspects of sales strategies concerning the selection of a result, which is actively recommended out of the presented result set. This describes a prime example for the usage of soft selections.

In the literature about consumer choice behavior all models treat the quality of the offered goods with respect to the customer's preferences as a major cognitive factor ([Nic66, EBK78, HS69, Han72]). Thus, the rules discussed in Definition 3.32 to Definition 3.36 that focus on good arguments regarding the preferences are also highly relevant for an intuitive and sales strategy driven query result presentation. But it is not always suitable to present the very best alternatives at first. Hansen ([Han72]) emphasizes that it depends on the situation, i.e. on the person involved, the location, and many other factors. Hansen also points out that it is very important for a customer to consider alternatives. It depends on the customer whether he wants to be presented with the best available alternative or wants to discover it by himself. Especially when opening a bargaining session in some situations it is smarter not to start with the best alternative, e.g. for having the possibility to qualitatively improve the offer during the negotiations ([Kni89, Shi97, Bir79]). Therefore, the following selection criterion might be useful, e.g. in order to start with an alternative of middle quality.

Definition 3.38 Presentation preferences regarding the overall quality

Assume a result set BMO* including $AQ_{overall}$. The presentation preference PP regarding a special desired overall quality within BMO* is defined as:

$$PP := AROUND(AQ_{overall}) \qquad *$$

A further very important selection criterion is the price according to the sales psychologist Bänsch ([Bän85]). He argues that on the one side recommending a result out of the visible result set with a high price shows the customer a high appreciation of his financial strength. But on the other side, this is embarrassing if the customer actually does not have this financial strength. Thus, the rule is when knowing about the highest price preference of a customer, then the result with the highest price and hence the semantically supposed item of highest product quality should be presented. If there is no knowledge about such a financial strength or the will to buy the most expensive alternative, then an alternative of a middle price up to a high price or even with the second highest price should be recommended. This shows appreciation of the customer, but it is not embarrassing for him when he cannot afford this product. And, in case of financial strength there is even a window of opportunity for the customer to step up to a product of higher quality and price, which boosts his ego. That leads to the following two selection criteria in Definition 3.39.

Definition 3.39 Highest or second highest price selection

Assume a result set BMO* including an attribute price. The presentation preference PP_h regarding the highest price is canonically defined as

$$PP_h := HIGHEST(price).$$

The presentation preference PP_{sh} regarding the second highest price can be defined as

$$PP_{sh} := HIGHEST(price)$$

over the relation $\{BMO^* \setminus \sigma[PP_h](BMO^*)\}$. *

In this section so far selection criteria were introduced for convincing the customer or for providing decision support to him in various situations. Moreover, the vendor may have his own preferences resulting out of BMO*, that he presents to support his utility function most, e.g. to increase his turnover. The following Definition 3.40 shows three typical vendor preferences supporting the business strategy of a vendor.

Definition 3.40 Vendor preferences within a presentation preference

Let price, profit, and purchase date be the corresponding attributes of BMO*. Then the vendor preferences for the presentation are for

a maximal turnover:	HIGHEST(price),
a maximal profit:	HIGHEST(profit),
promoting slow sellers:	LOWEST(purchase date). *

Yet, these vendor preferences/utility functions ([WD02]) do nothing to convince the customer or to support his decision process. Naturally, good arguments to convince a customer are products that satisfy his preferences. But all selection criteria were intuitively formulated as wishes, i.e. as soft conditions over the result set. Thus, all presentation preferences introduced can be applied in combination, where conflicts within the aggregated preferences can easily be handled due to the flexible underlying theory of [Kie02, Kie04].

Example 3.30 Aggregated presentation preferences in a sales scenario

Continuing Example 3.21, where Marge queries Apu's notebook database, he wants to point out the results with the most 'perfect' single sales arguments. If possible, without adverse effect he wants to support his turnover per piece (see Definition 3.40). I.e. Apu's presentation preference reads as:

PP := HIGHEST($AQ_{\text{'perfect'}}$) & HIGHEST(price)

With reference to Table 3.12 and Table 3.13 and Figure 3.7 result t_1 and t_3 both have the most 'perfect' sales arguments, i.e. six. Apu will present t_1, because within these two results t_1 supports his turnover better. With the gathered information Apu can now present the six best matching results and can point out the one reflecting his presentation preference. He argues as follows, where he decides for a blatant argumentation:

> *"There are **six** best matches with respect to your preferences. I recommend the Toshiba notebook position. It **perfectly** matches your desired manufacturer **and perfectly** fulfills your preferences regarding to warranty period, home repair service, chip set, **and** the CPU **and** main memory make. This is a **good** choice."*

Apu only uses the 'perfect' arguments without mentioning some negative aspects. For the credibility he adds the statement about the 'good' overall quality.

If Apu considers single sales arguments equally important to a high turnover then the following presentation preference describes the sales strategy:

PP := HIGHEST($AQ_{\text{'perfect'}}$) $\otimes$ HIGHEST(price)

Please note in [Kie04] is shown that this strategy can not be expressed by a preference model based on ranking. ⌂

3.7 Valuating results of other search technologies

In this chapter a powerful framework has been developed so far for valuating search results. Filter criteria applied to the result set have been discussed and selection criteria have been defined according to general presentation rules and rules designed especially for a sales strategy driven presentation. The customer-friendly and intuitive BMO search of [Kie02, Kie04] was used as the initial position, because this approach does not bother the customer with irrelevant results dominated by other ones and delivers all relevant results at once. In this last section it is pointed out that the introduced presentation framework is not limited to a BMO search result set. Indeed, every imaginable result set could be used as input within this framework, e.g. the result set of a so called 'top-k' search as introduced in [BCG02]. This framework is universal, because it satisfies the postulate of Definition 3.3 and because the valuation itself is based on the flexible, intuitive and semantics based preference model of [Kie02, Kie04]. This is briefly discussed in the following example.

Example 3.31 Preference based presentation on an arbitrary search result

Homer wants to buy cheese at Apu's online store. He **prefers** cheddar cheese with a fat content of **around** 55%. **Equally important,** the state of production **should** be California. Apu's search engine has computed results t_1 - t_3 for Homer's preference

$$P := \text{POS(Sort, \{Cheddar\})} \otimes \text{AROUND(Fat content, 55)} \otimes \text{POS(State, \{California\})},$$

as shown in Table 3.15, following the presentation rule of a presentation style guide to present at least three alternatives to a customer.

Sort	Cheddar	Cheddar	Camembert
Fat content	45	10	50
State	California	Oregon	California
	t_1	t_2	t_3

Table 3.15 Sample arbitrary search result set

Without describing the details, the quality valuation is assumed to deliver the following overall qualities:

- $QUAL_{P,s}(t_1)$:= 'very good'
- $QUAL_{P,s}(t_2)$:= 'acceptable'
- $QUAL_{P,s}(t_3)$:= 'good'

Homer senses the quality valuations as appropriate, but does not really understand why Apu also put t_2 on the result table, because t_2 is dominated by t_1. ⌂

Presenting dominated results is not senseless in general as discussed in [Han72]. If there are only few results, it is often a successful strategy to additionally present some bad results to the customer. He might be convinced of the best matching alternatives, because obviously other available results are even worse. But as seen in this example, the design of the introduced presentation framework also intuitively supports other result sets than the canonical BMO.

The theory has been implemented and is in the following named as the Preference Presenter technology. Summarizing, with its help search results can be valuated according to each involved preference. Quality claims and other filter criteria can be applied. Out of the result set, one or more results can be pointed out that are preferred for a proactive presentation. Presentation preferences for this purpose can be flexibly combined to support a special presentation strategy. In the next chapter it is shown that this preference based presentation framework plays a major role in an automated offer composition within the e-procurement sales process.

4 Personalized Offer Composition for E-Procurement

E-procurement today is a very lengthy and costly process. Scrolling through huge electronic product catalogs is still state-of-the-art. In this chapter the process of purchasing goods is analyzed regarding an improved, automated electronic procurement process. In the second section personalized and preference based technologies are composed to realize this automated process for an effective e-procurement. There, two novel frameworks for a personalized price fixing are introduced. Following this approach, data integration aspects are discussed, under consideration of existing IT standards, for easily integrating data from product databases and customer relationship management systems ([RL02]).

4.1 E-procurement - state-of-the-art

This section outlines the workflow of an e-procurement process for searching and purchasing products that support a fully automated offer composition. Afterwards, current problems of state-of-the-art applications are discussed.

4.1.1 Automated e-procurement sales process

As described in [ÖFA01] an e-procurement process for the customer consists of four steps, i.e. searching through catalogs for desired products, pricing and ordering, delivery, and payment and controlling. Yet, some parts are still done manually. To build up an automated offer composition in e-procurement applications the workflow of the first two steps is illustrated in Figure 4.1. Firstly, the shopping cart is filled step by step with desired products and corresponding quantities. This process includes the product search, a sales strategy based presentation of the search results, and the decision, whether to put one or more results into the shopping cart, eventually. Secondly, the price for the shopping cart as a whole is determined. According to his practice the customer bargains about the price. At last, the customer has the choice whether to accept the offer or to get an open offer valid for a specific period. Of course, the individual steps in that process are not bound to a linear sequence of actions. It is possible and reasonable to reiterate some steps, e.g. to change the shopping cart after the price offer. For an effective and efficient sales process at the product presentation as well as at the price fixing, i.e. at the price offer and during the bargaining, cross selling, up selling, and down selling are very important sales techniques. Detailed information and how to apply them are described in economy literature ([Dom02]). Please note that cross/up and down selling can also have an impact on the correlated price of a product (bundle). This aspect is separately handled in this thesis.

Definition 4.1 Cross selling

Cross selling means to proactively offer additional products AP which are in relation to the current products of concern CP. A cross selling attempt of the vendor is defined as cross_selling(CP, AP) $\mapsto$ PB,

where PB is the new product bundle, with CP $\subseteq$ PB. *

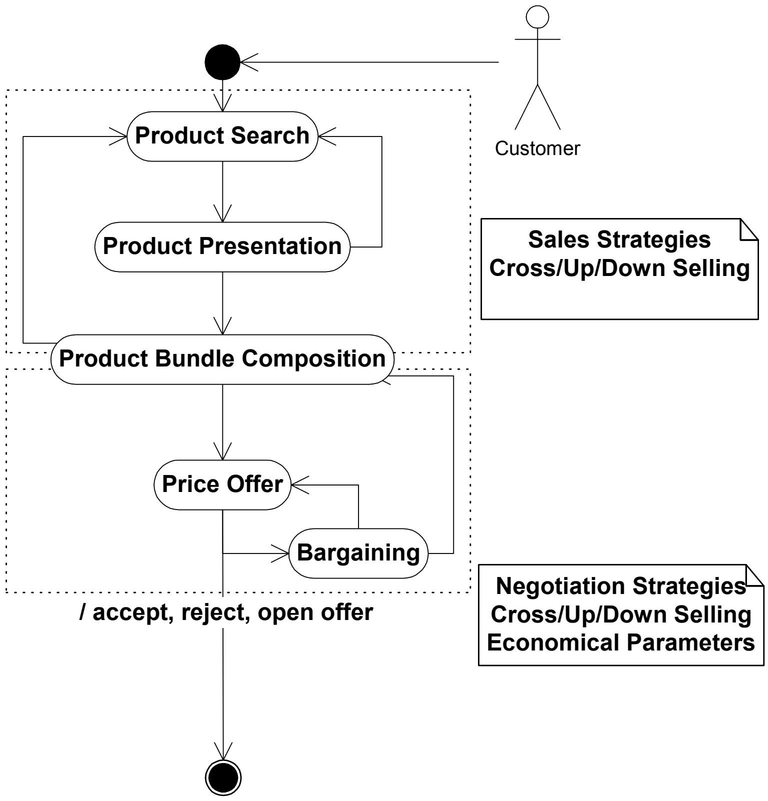


Figure 4.1 Workflow of an automatic offer composition

Example 4.1 Cross selling

Homer asks Apu's web store for notebooks and "Dolby Surround" loudspeakers. Beside the desired notebooks and loudspeakers, Apu's web shop also proactively offers fitting carry cases for the notebooks, three-year warranty packages for the notebooks, and also the necessary sound cards for the advanced loudspeakers. Homer accepts all but the sound cards. So, Apu's cross selling attempt is written as

cross_selling({notebooks, loudspeakers}, {carry cases, warranty packages, sound cards}) =
{notebooks, loudspeakers, carry cases, warranty packages}

with the new shopping cart content of concern. ⌂

Thus, cross selling can increase the turnover but can also improve the customer satisfaction, because it is a helpful service to recommend useful accessories or even mandatory additional products. Down selling and up selling are defined for the more common case where only one product is in the focus, but of course can also be applied to product bundles.

Definition 4.2 Down selling

Down selling means to proactively offer a cheaper product dp instead of the product cp, which is the product of current concern, to safe the sale or to improve the customer satisfaction. A down selling attempt of the vendor is defined as

$$\text{down\_selling}(cp, dp) \mapsto p,$$

where p is the product of concern after this down selling attempt with $p \in \{cp, dp\}$ and $price_p \leq price_{cp}$. *

Example 4.2 Down selling

Homer asks Apu's web store for Hewlett Packard notebooks. But the price for these high-quality notebooks is far above the clouds for Homer's financial scope. So, as an alternative Apu's shop offers notebooks from Acer. Homer is convinced of this cheaper alternative. This down selling attempt is written as

down_selling(Hewlett Packard notebooks, Acer notebooks) = Acer notebooks

with the new notebooks of concern. ⌂

Definition 4.3 Up selling

Up selling is to proactively offer a more expensive product up instead of the product cp, that is currently considered by the customer, in order to increase the turnover. An up selling attempt of the vendor is defined as

$$\text{up\_selling}(cp, up) \mapsto p,$$

where p is the product of concern after this up selling attempt with $p \in \{cp, up\}$ and $price_p \geq price_{cp}$. *

Example 4.3 Up selling

Marge asks Apu's web store for very cheap Sony notebooks. But Apu knows about the financial strength of Marge and her tendency to high-quality goods. During the bargaining he proactively offers notebooks from Hewlett Packard and persuades Marge with his arguments about the quality. This up selling attempt is written as

up_selling(Sony notebooks, Hewlett Packard notebooks) = Hewlett Packard notebooks. ⌂

4.1.2 Deficiencies of state-of-the-art technology

The last section has shown the necessary steps for a sales automation. Yet, there are still enormous problems for a realization of this process with state-of-the-art technology that are discussed in the following three categories.

1. Product search

The misery starts with the product search. Customers are mostly still used to scroll manually through huge electronic product catalogs, which is very intensive in terms of time and costs. A better way would of course be to use a search engine in combination with an intuitive and customer-friendly interface, like many B2C shops do. But the problem in B2B is the complexity and the variety of characteristics of the products. Thus, state-of-the-art approaches to find desired products are not enough for the customer ([XAF03]). Today's search engines interpret the customer's search conditions as hard constraints and often confront him with the empty result effect. When interpreting the constraints as 'or'-conditions, the flooding effect with irrelevant results occurs. A further approach is to iteratively ask the customer to soften his search criteria, which is a very frustrating and time-intensive process. There is the need for a search engine that delivers best alternatives when there is no perfect match. A preference search engine as described in section 2.2 solves this problem.

2. Smart sales approaches

Unlike human sales agents, e-business applications do not consider principles of sales psychology when presenting the search results. In the real world a sales agent has to find a way to satisfy his own preferences and the preferences of the customer as well as possible, which is a challenging act. To convince the customer of the goods offered a *major factor is to argue about the quality of the presented products with respect to the search preferences*. This is stated by well-known models of customer choice behavior ([Nic66, HS69, EBK78, Han72]) within sales scenarios. Some of today's search engines compute alternatives in case of a missing perfect match, but they are not able to provide semantic information about the search result quality. This information can be provided by the Preference Presenter. Moreover, there is no information about product coherences like mandatory articles fitting selected products. As a good service, the customer may normally expect to get additional information from an attentive vendor whether there are useful accessories to the concerning products. For example, often a printer cable is missed after buying a printer. But besides cross selling a good e-procurement portal should even be able to use up and down selling techniques, e.g. when it recognizes that the products are too expensive for the customer, in order to safe a sale. For a *deep personalization*, there must be known whether a customer is sensitive for e.g. cross selling. This information can be managed in a Preference Repository.

3. Price fixing

Current e-procurement portals either are not able to tell a price for the products of the shopping cart or only summarize list prices for the concerning goods. A customer expects to be individually treated according to his price preferences and general conditions, e.g. net price freight included. And of course, he expects personalized discounts, according to his buying position and behavior. E.g. a customer who often orders blue clear stock boxes expects a higher discount for this article.

Also this can be managed in a Preference Repository. Depending on the customer's buying pattern, an advanced e-procurement portal should provide the opportunity for further price discussions.

4.2 Preference based components

Advanced e-commerce applications require a high level of personalization and must be able to react to different situations. This section presents a novel approach how to effectively realize the business process of Figure 4.1 while solving the problems of section 4.1.2 by composing preference based components.

4.2.1 Technology for a preference based and smart product composition

The iterative product search and result presentation that leads to a product bundle composition as described in the upper box of Figure 4.1 is a decisive procedure. A search technology that respects customer preferences and therefore avoids the first deficiencies described in section 4.1.2, was described in section 2.2. As illustrated in Figure 4.2, where the interplay of preference technology for the product bundle composition is shown, alternatively Preference SQL or Preference XPath respect the customer's present search preferences as well as his long-term search preferences, and the vendor's preferences. The BMO result set is delivered to the Preference Presenter.

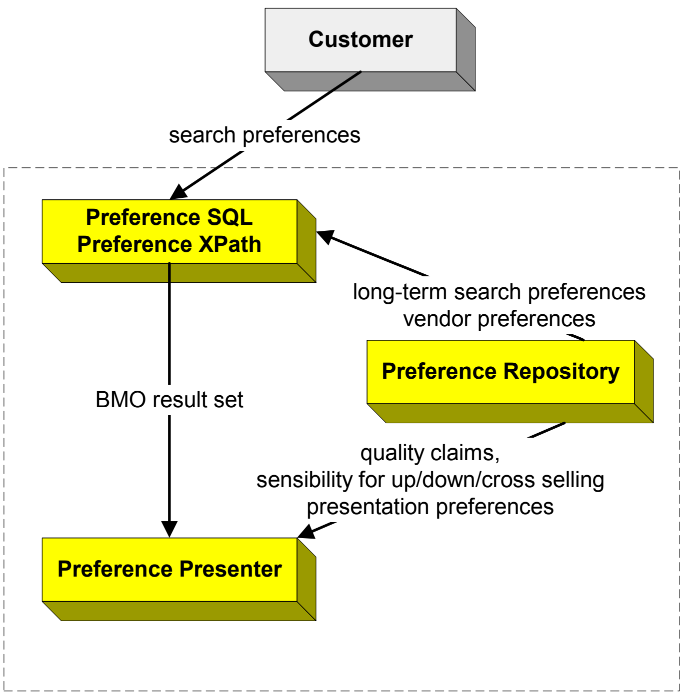


Figure 4.2 Preference based composition of the shopping cart

How to smartly present the results is extensively elaborated in chapter 3, including the feature to apply different sales strategies. In this, the presentation preferences which represent major aspects of the sales strategies are arranged in the Preference Repository with its situational context. The Preference Presenter also gains information about the quality claims of the customer as well as further vendor dependent ‘but-only’ filter criteria from the Preference Repository. Furthermore, for different situations the sensibility for up selling, down selling, or cross selling of a customer during the product bundle composition is individually stored. E.g. a predestined moment for a cross selling attempt is when a customer adds presented products to his shopping cart. Therefore, this personalized approach is able to cope with the second set of problems in section 4.1.2. The compatibility of the composed technologies shown in Figure 4.2 is given, since they all base on the same intuitive preference model of [Kie02, Kie04].

For a standardized and comfortable data interchange of the quality information between platforms or components an appropriate design is an XML based structure. In Figure 4.3 the quality information of the result tuple t_1 of Example 3.30 is given. At this, all quality valuations of all base, Pareto, and prioritized preferences involved are given, as well as the distances of the numerical values from the optimum and the levels within a categorical preference.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PREFERENCEQUALITY (View Source for full doctype...)>
<PREFERENCEQUALITY>
 <BMORESULTSET>
  <RESULT id="13324" favorite="true">
   <QUAL type="complex" subtype="prioritized" characteristic="notebook" value="good">
    <QUAL type="complex" subtype="pareto" characteristic="priority1" level="1" value="good">
     <QUAL type="base" subtype="posneg" characteristic="notebook make" value="perfect" />
     <QUAL type="complex" subtype="prioritized" characteristic="main memory" value="good">
      <QUAL type="base" subtype="highest" characteristic="main memory capacity" value="good" level="1" distance="512" />
      <QUAL type="base" subtype="posneg" characteristic="main memory make" value="perfect" level="2" />
     </QUAL>
     <QUAL type="complex" subtype="prioritized" characteristic="cpu" value="perfect">
      <QUAL type="base" subtype="pospos" characteristic="cpu make" value="perfect" level="1" />
      <QUAL type="base" subtype="around" characteristic="cpu clock frequency" value="good" level="2" distance="0.4" />
      <QUAL type="base" subtype="pos" characteristic="cpu chip set" value="perfect" level="3" />
     </QUAL>
     <QUAL type="base" subtype="lowest" characteristic="price" value="sufficient" distance="2049" />
    </QUAL>
    <QUAL type="complex" subtype="pareto" characteristic="priority2" value="good" level="2">
     <QUAL type="base" subtype="lowest" characteristic="weight" value="acceptable" distance="1.9" />
     <QUAL type="complex" subtype="prioritized" characteristic="warranty" value="perfect">
      <QUAL type="base" subtype="around" characteristic="period" value="perfect" level="1" distance="0" />
      <QUAL type="base" subtype="pos" characteristic="home repair" value="perfect" level="2" />
     </QUAL>
     <QUAL type="complex" subtype="prioritized" characteristic="delivery terms" value="good">
      <QUAL type="base" subtype="lowest" characteristic="time" value="good" level="1" distance="0" />
      <QUAL type="base" subtype="around" characteristic="order quantity" value="very good" level="2" distance="10" />
     </QUAL>
    </QUAL>
   </QUAL>
  </RESULT>
 </BMORESULTSET>
</PREFERENCEQUALITY>
```

Figure 4.3 Sample preference quality XML file for one BMO search result

Thus, arguments easily can be developed e.g. by selecting the information of the ‘order quantity’ base preference. Because of the given subtype ‘around’, the valuation ‘very good’, and the distance of ‘10’ the argument is clear, namely the order quantity is only 10 pieces away from the optimum and therefore a very good alternative. Due to reasons of clarity some meta information like direction of the distance and the measuring unit were disregarded in Figure 4.3. The complete DTD for this structure is added in Appendix A.

4.2.2 Personalized price offer

In e-procurement for each customer personalized discounts are taken into consideration for each product. The basic price to which to apply the personalized product discounts is the so called list price. If a customer orders a higher number of one good then the price can decrease stepwise by a so called differential price. After applying personalized product discounts, discounts for the whole requested product bundle are deducted, e.g. a volume discount. This information is normally stored in the so called customer relationship management ([RL02]). The difficulties of state-of-the-art technology at this point toil in exporting and integrating this information into an e-procurement portal. A solution for the data export of the product discounts and the differential prices, respectively, as well as the list prices is meanwhile given by catalog standards like BMEcat. Their application is discussed in section 4.2.4. Due to missing standards, as discussed by business data processing specialists ([KLS02]), in this section an XML based structure is modeled in order to handle each kind of discount as a first standardization approach. Then, an algorithm is introduced that calculates the personalized price to offer to the customer under taking into consideration all product discounts and product bundle discounts.

In the economy literature ([WD02]) there are lots of names and semantics for discounts, which are used in different ways, e.g. sequential computation or summed computation. Regarding to product bundle discounts there are for example the following:

- ♦ volume discount
- ♦ season discount
- ♦ personal customer discount
- ♦ branch discount
- ♦ currency discount
- ♦ cash discount
- ♦ advertising discount
- ♦ online order discount
- ♦ trading discount
- ♦ regular customer discount

Product discounts generally are deducted first and a cash discount last. But there are no general rules from which calculation basis, in which order the other discounts to deduct from, respectively. E.g. a vendor can grant a branch discount of 5% and a 3% online discount. But the discounts have different effects depending on whether they are deducted in succession or as 8% at once. With several discounts the calculation basis, starting with the summed turnover of the list prices minus the product discounts and differential prices, respectively, can change several times. Moreover, most of these discounts are tied to conditions, e.g. a volume discount requires a special volume of the turnover, e.g. 1000 € or more.

Definition 4.4 Discount and its necessary parameters

A discount d with its necessary information is defined as

d(n, cl, v, vu, vt, c, cp),

where

- n is the name of the discount,
- cl is the calculation level ($\geq 0, \in \mathbb{N}$),
- v is the value of the discount,
- vu is the unit of the value v, e.g. € or %,
- vt is the value type, i.e. absolute, relative, or a user defined condition,
- c is the condition which is decisive for applying the discount, and
- cp are condition parameters, necessary for checking whether c is fulfilled or not. *

In the following Algorithm 4.1 a personalized price offer is computed, taking all mentioned discounts into consideration.

Algorithm 4.1 Personalized price offer

Let t be the turnover of a product bundle, calculated as the sum of the net list prices of the products of the bundle. The total net amount for an offer with a given set of discounts $D := \{d_i \mid i = 1, \ldots, n\}$, d_i defined according to Definition 4.4, is calculated as follows:

calculation_basis := t; levelcount := 0;
calculation_basis := calculation_basis - all differential product discounts;
while (D != empty) **do**
  calculate the savings of all discounts d_i(n, cl, v, vu, vt, c, cp) where cl == levelcount based on calculation_basis; let s be the sum of these savings;
  calculation_basis := calculation_basis - s;
  D := D \ {d_i(n, cl, v, vu, vt, c, cp) | cl == levelcount};
  levelcount := levelcount + 1;
end while
return calculation_basis; *

Example 4.4 Personalized discounts for an audio CD reseller

Bart wants to sell music CDs and wants to procure them at Apu's web store for resellers. After the product composition and deduction of (differential) product discounts the product bundle of Table 4.1 is of concern.

Performer	Title	Quantity	differential net price/piece
Peter Gabriel	Up	250	5.60 €
Mike Oldfield	Crises	85	3.55 €
Lionel Richie	Encore	350	5.45 €

Table 4.1 Sample product bundle of audio CDs

Apu's web store considers the discounts illustrated in Figure 4.4 organized in the newly introduced XML based structure of Appendix B. Apu knows about Bart's practice to immediately pay by credit card and also that Bart picks up the goods at Apu's entrepot. And because Apu wants to support orders at weekend he grants 5% for Bart's order at Sunday. The calculation using Algorithm 4.1 is obvious.

```
…
<DISCOUNT NAME="WeekendDiscount" LEVEL="0" VALUE="5" VALUEUNIT="%"
    VALUETYPE="relative" CONDITION="current day is weekend day"/>
<DISCOUNT NAME="CardDiscount" LEVEL="1" VALUE="3" VALUEUNIT="%"
    VALUETYPE="relative" CONDITION="immediately payment by card"/>
<DISCOUNT NAME="PickupDiscount" LEVEL="1" VALUE="10" VALUEUNIT="%"
    VALUETYPE="relative" CONDITION="no delivery of goods"/>
…
```

Figure 4.4 Sample discounts as flexible XML based structure

⌂

For a first standardization approach in Appendix B a newly modeled DTD structure is given, carrying the information of Definition 4.4. An example was given with Figure 4.4. This XML based structure can easily be used for a platform independent electronic data interchange (EDI [Jil94]) and for example directly be integrated into "user_defined_extensions" of customer description standards like xCIL, the eXtensible Customer Information Language, or xCRL, the eXtensible Customer Relationship Language ([Kum02]).

So, a way was shown how to apply personalized discounts of various situations in a standardized manner. In Figure 4.5 the schematic interplay to solve the third problem of section 4.1.2 is illustrated. The necessary situational information about preferred practices and discounts involved are organized in the Preference Repository, e.g. the practice to immediately pay via credit card. In which style the presentment should be done is also organized in the Preference Repository, e.g. a customer wants to have his price offers as gross price, or a customer wants all discounts itemized, another customer only wants to see a sum of discounts.

For solving the third problem of section 4.1.2 there should be the possibility to bargain. Also all preferences and their situational context for bargaining are arranged in the Preference Repository, namely the sensibilities for cross/up/down selling in bargaining situations and the preferred negotiation strategies personalized for each customer. The model of the Preference Bargainer is introduced in the following section 4.2.3.

4.2.3 Preference Bargainer

Bargaining is a very old and traditional way of price fixing, which can also have a very positive effect in e-commerce ([LD99]). For the basic rules of bargaining see [Rai90]. With the Preference Bargainer for the first time an electronic bargainer with the ability for multi-objective negotiations is introduced. Because this part of the thesis has already been published under [FKHF02], beside a short summary for the sake of completeness, in this section an algorithm is given for multi-objective bargaining.

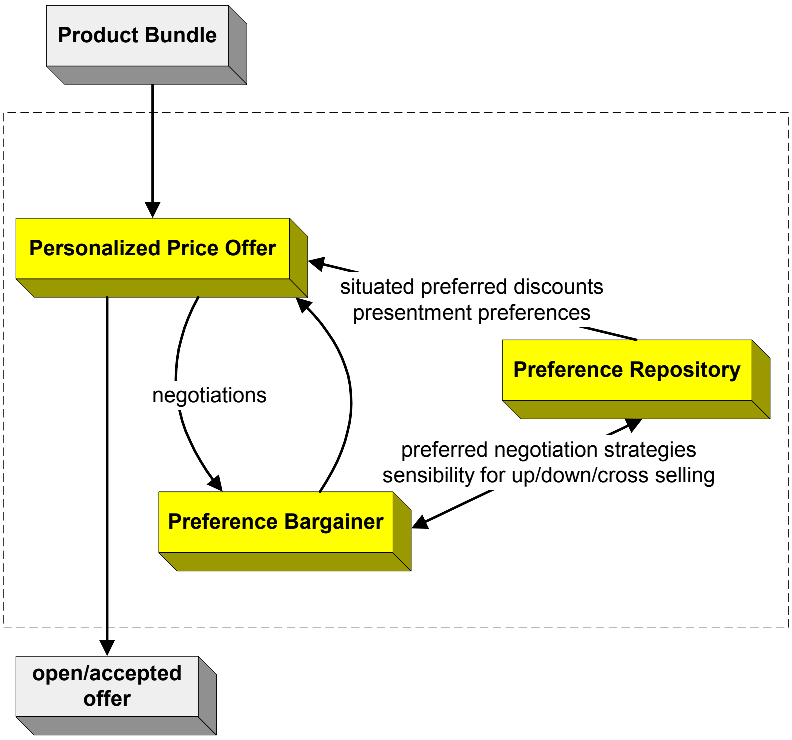


Figure 4.5 Personalized and preference based price fixing

Definition 4.5 Bargaining session and bids

Let PB_i^N be the product bundle of concern by N ∈ {V, C}, which denotes the vendor and the customer, at the time i. Let t_i^N be the turnover, i.e. the current offered price at the time i offered by N. A bargaining session is a sequence

$b_1, \ldots, b_n$

of bargaining steps called bid pairs, where each bid pair b_i is of the form

$b_i := (b_i^V , b_i^C)$, where

$b_i^V := (t_i^V , PB_i^V)$ is the vendor's price offer for PB_i^V, and

$b_i^C := (t_i^C , PB_i^C)$ is the customer's bid. *

The Preference Bargainer of course works profitably, is smart and unpredictable, because it works with various negotiation strategies like linear, progressive, 'tit for tat', 'doing the opposite' or random strategies, and adapts to the customer's needs. Moreover, the Preference Bargainer works personalized for each customer and not only bargains about the price. This multi-objective approach of an electronic bargainer proactively offers additional products and services that fit the products of concern if the customer is sensitive for this in the specific situation. To safe a sale if a situation requires a cheaper offer the Preference Bargainer uses down selling techniques. If it recognizes a financial strength of the customer and knows about the tendency for high quality goods the Preference Bargainer searches by respecting the customer's preferences for more expensive goods and makes an up selling attempt. In Algorithm 4.2 the application flow from the vendor's view is specified.

Algorithm 4.2 Multi-objective bargaining

Let t_i^N, PB_i^N and b_i be defined according to Definition 4.5. Cross/up/down selling attempts are defined according to Definition 4.1 to Definition 4.3. Let cr_i be the *customer reaction* to a bid of the vendor, which can be *acceptance, rejection, or a further bid*. And let ecb_i be the result of the *evaluation of the customer behavior* after b_i, which is the vendor's *rejection, acceptance, or a further bid* with an *optional up/cross/down selling* attempt. Then a multi-objective bargaining session from the vendor's view works as follows:

stop := false; i := 1;
while (!stop) **do**
    calculate and make bid b_i^V;
    wait for cr_i;
    if (cr_i == accept) **then return** accepted offer with turnover t_i^V and PB_i^V;
    if (cr_i == reject) **then return** open offer with turnover t_i^V and PB_i^V;
    calculate ecb_i;
    if (ecb_i == accept $\vee$ reject) **then** stop = true; break;
    i++;
    switch (optional)
        case == "up selling" **do** $PB_i^V = (PB_{i-1}^V \setminus CP) \cup$ up_selling(CP, UP);
        case == "down selling" **do** $PB_i^V = (PB_{i-1}^V \setminus CP) \cup$ down_selling(CP, DP);
        case == "cross selling" **do** $PB_i^V = PB_{i-1}^V \cup$ cross_selling(CP, AP);
    end switch
end while
if (ecb_i == accept) **then return** accepted offer with turnover t_i^C and PB_i^C;
else return open offer with turnover t_i^V and PB_i^V; *

Please note that, even if there is no agreement the customer always gets an open offer. In this section 4.2 so far technology components were modeled or combined to enable an effective and efficient personalized e-procurement sales process. Beside of preference knowledge stored in the Preference Repository lots of economical data must be integrated. Therefore, this section is completed by the following data integration aspects.

4.2.4 Data integration by means of e-procurement standards

A very critical issue is the standardized data integration into an e-procurement portal. Various mostly XML based standards for describing products have arisen during the last years ([EZ02]), e.g. cXML[12], ICE[13], RosettaNet[14], BizTalk[15], and BMEcat[16]. By describing product characteristics most extensively the XML based standard BMEcat is predestined for data interchange and integration. As briefly shown in Figure 4.6 there is a lot of semantic information within the article description. Therefore, BMEcat ([SKPRH01]) by now is widely used for electronic data interchange ([Jil94]) between and also within enterprises.

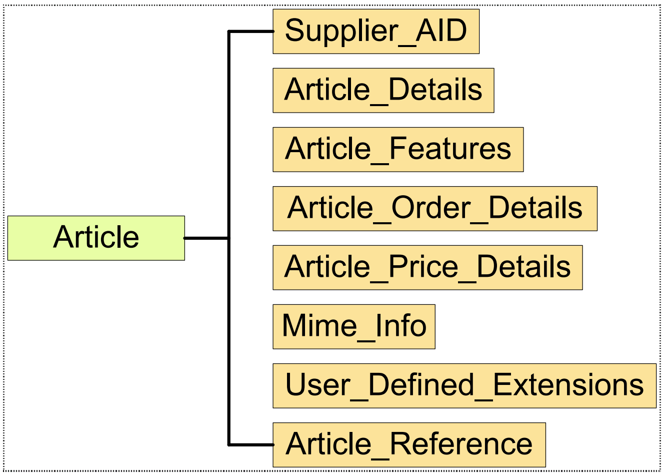


Figure 4.6 Structure of the BMEcat tag <ARTICLE>

Some aspects of BMEcat are briefly considered in more detail. Besides information about e.g. units per package, net or gross prices, and freight included or not, with the Article_Reference XML tag several highly relevant semantic information regarding several issues is provided. In this tag interdependencies between two products are described, namely the following types. The referenced article can be

- a spare part of the article,
- similar to the article (namely a suitable alternative),
- the follow-up model,
- a mandatory additional article,
- the same product in a different packaging,
- an accessory, or
- part of the currently described article.

If a customer is looking for a special product that is listed in the product selection but temporarily not or no longer available then a natural alternative set of products could be gained out of the type attribute of the Article_Reference XML tag. So, for the product search automatically a POS/POS preference can be constructed and considered, i.e. a set of alternatives with the products of the ref-

[12] http://www.cxml.org
[13] http://www.w3.org/TR/NOTE-ice
[14] http://www.rosettanet.org
[15] http://www.microsoft.com/biztalk/

erence type 'similar', 'follow-up', and 'different packaging', because all these products have very similar characteristics as the preferred product. Of course the same information can be used to search for down selling and up selling candidates. Further interdependencies are illustrated in Figure 4.7. Obviously, article references of type 'spare part' or 'accessory' are predestined for a cross selling attempt.

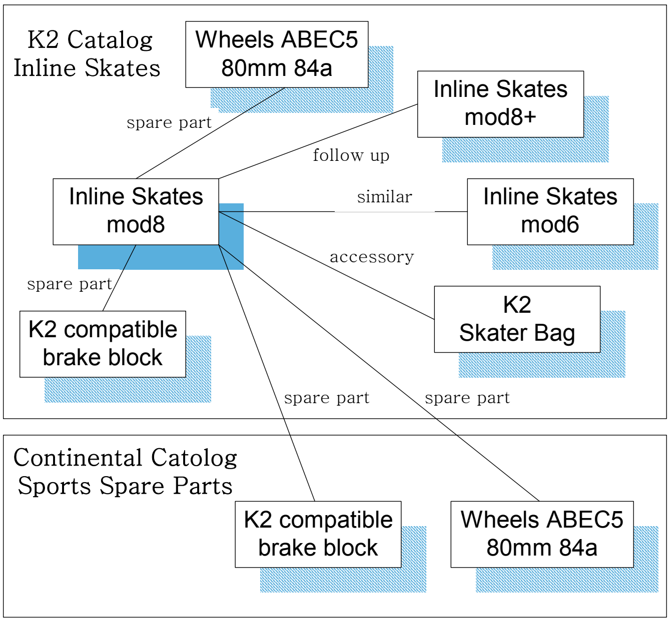


Figure 4.7 Example of BMEcat Article_Reference interdependencies

To describe the very product dependent characteristics in a standardized manner, different organizations try to formulate feature standards for special product groups. These feature standards can easily be integrated into BMEcat. The most popular standards are proficl@ass[17], ETIM[18], UNSPSC[19], and eCl@ss[20]. The latter standard is often used in combination with BMEcat. eCl@ss ([Pal00]) uses a 4-tier key to classify products, where each tier of the key consists of two digits. The key is hierarchically ordered and nested as follows:

1. tier: subject group (e.g. 21; tools, machine tools)
2. tier: main group (e.g. 21-10; works equipment)
3. tier: group (e.g. 21-10-04; warehouse equipment)
4. tier: subgroup (e.g. 21-10-04-21; clear stock box)

In contrast to a preference search, a parametric search ([AW00]) would require at least four steps. With the eCl@ss key the kind of a product is uniquely determined. But of course, there are differences between for example clear stock boxes. And therefore, for each classified product there is an additional list of standardized features. Following the clear stock box example, some of the features are manufacturer, weight, height, length, and material (see Figure 4.8). This categorized semantic information can be used when interpreting a preference search and result presentation, in contrast to e.g. a simple full text search.

[16] http://www.bmecat.org
[17] http://www.proficlass.de
[18] http://www.etim.de
[19] http://www.un-spsc.net
[20] http://www.eclass.de

Classification	Description
21-10-04-21 [AAA619001]	Clear stock box
Keywords:	Euro-fix box, Stock box,
Attribute-Set:	CAA074001- Color CBA003001- Conditions of delivery CBA002001- Dimension standard AAA889001- EAN/UCCCode CBE001001- Height CBA005001- Identification CBA007001- Length AAA001001- Manufacturer AAA252001- Manufacturer article number CBA102001- Manufacturer's product name CBA103001- Manufacturer, id number CBA001001- Material, short name CBA101001- Product description CBA004001- Quality features, certificate CBB018001- Weight CAB021001- Width

Figure 4.8 eCl@ss characteristics for clear stock boxes

In terms of integrating economical data into an e-procurement portal list prices can be imported with BMEcat. Moreover, by using the price_update catalog, a short form of BMEcat, personalized product discounts and personalized differential product prices can easily be integrated. Together with the data interchange structure for product bundle discounts (Appendix B) an automatically computed price can be offered as illustrated in Figure 4.9. In case of bargaining the necessary personalized knowledge about the total price limits can also be exchanged via price_update BMEcat catalogs.

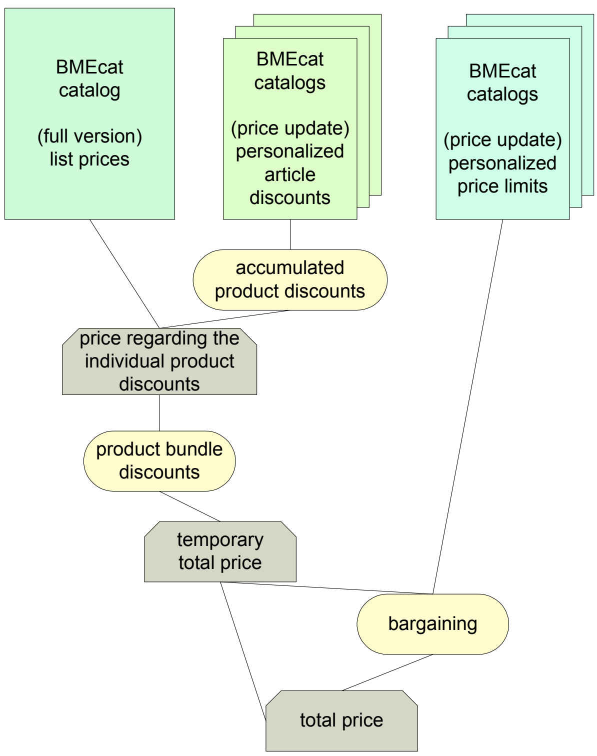


Figure 4.9 Price fixing by means of BMEcat data import

5 Automated E-Procurement Sales Agent

COSIMA^{B2B}, an automated sales agent for e-procurement is the consequent realization of the e-procurement framework elaborated in chapter 4 under usage of the novel Preference Presenter technology presented in chapter 3. Firstly, a short history of the virtual assistant COSIMA is given, followed by a description of the functionality of the automated sales agent COSIMA^{B2B}, including a short shopping tour. Afterwards, the personalization manager, an administration tool for managing lots of personalization parameters, is introduced. Finally, a technical evaluation of the integrated novel Preference Presenter and the Preference Bargainer as well as sophisticated results of experimental settings with different test customers are presented at the end of this section. This prototype gives evidence that a complex sales process can effectively be done by an automated e-sales agent.

5.1 History

The COSIMA project, aiming at the development of preference based services for a more human-like and intuitive online shopping process, started in the beginning of the year 2000. The first COSIMA prototype was a comparison shop in combination with a female human-like embodied character agent named COSIMA and dynamic speech synthesis. This work was presented at the WECWIS 2001 ([KFHE01]) and as prototype demonstrated at the SIGMOD 2001 ([KHFE01]). With online search agents COSIMA searched internet shops in the fields of books, audio CDs, and computer hardware respecting the customer's search preferences. About 600 test customers installed this freely available service. Moreover, COSIMA was presented to a large audience at the computer fair SYSTEMS[21] 2000.

Further research in the B2C area led to the multi-objective bargaining component (first version of the Preference Bargainer) which was introduced as COSIMA2 at the AAMAS 2002 ([FKHF02]), and presented at the worlds largest computer fair CeBIT[22] in 2002. COSIMA2 searched for best matches in the computer hardware domain and bargained not only about the price. Additionally, she used up/down and cross selling techniques and offered e.g. additional services like extended warranty, express delivery, or home repair service. Thereby, COSIMA2 learned about the customer's behavior and adjusted to the customer's preferences.

Parallel work was the design and implementation of a first autonomous buying agent, the Preference Agent ([Rei02]). The P-Agent needed a budget, a time period, and a list of wishes. With that the P-Agent purchased the desired goods of the customer by searching different web shops and auction platforms following different strategies and respecting the customer's preferences.

[21] http://www.systems-world.com
[22] http://www.cebit-world.com

With these works there came numerous incentives for a deployment into the fields of B2B. After the COSIMA project joined the Bavarian research association FORSIP[23] in the middle of 2002, the bargaining component was enhanced for the usage in B2B applications and redesigned as a middleware component. The work resulted in the development of the COSIMA^{B2B} prototype, a deeply personalized sales agent for the complex offer composition process within e-procurement. Such a flexible and complex enterprise system needs a solid and flexible platform. There component based approaches ([GT00]) like J2EE application server systems ([HL03]) are recommended. Implementations of the J2EE standard are for example IBM Websphere[24], BEA Weblogic[25], Oracle Application Server, and the open source solution JBoss Application Server[26] that run on almost every operating system. For an overview of the flexible architecture of the COSIMA application server and its components see Figure 5.1. The source code for these technologies by now counts more than 100.000 lines.

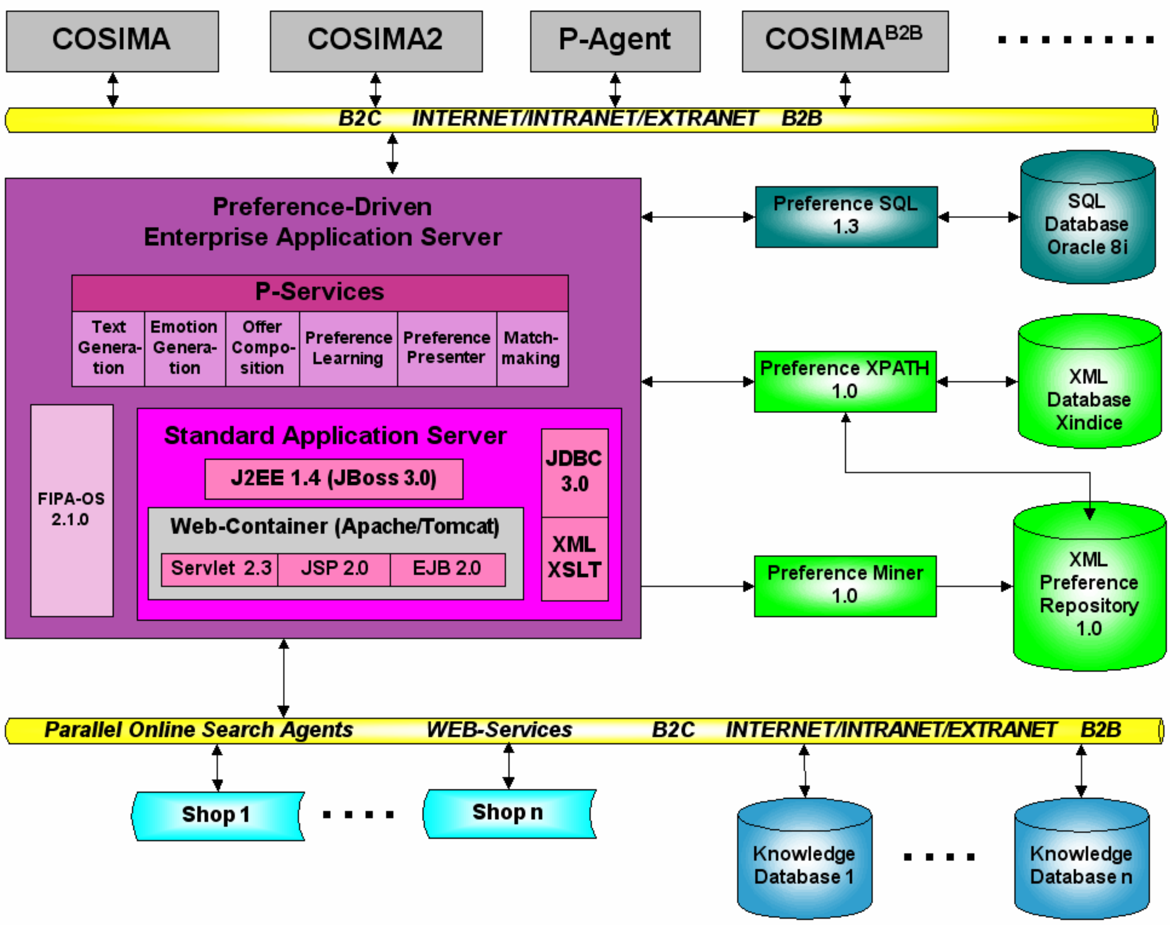


Figure 5.1 Architecture of the COSIMA application server

---

23 http://www.forsip.de
24 http://www.websphere.com
25 http://www.bea.com
26 http://www.jboss.org

5.2 The prototype COSIMA^{B2B}

COSIMA^{B2B} is the prototype of an autonomous sales agent that automates a cost intensive e-procurement process, consequently realizing the model and suggestions of chapter 4. Together with the industrial partners SSI Schäfer[27] (seller-side), MAN Roland Druckmaschinen AG[28] (buyer-side), and Fachverlag Walch[29] (content provider) a typical B2B use case was modeled. The product domain comprises boxes, in particular storage, transport, and waste containers according to the domain of the industry partners. Insertion and adaptation of the industry partner's product catalog is easily achievable using the XML based BMEcat standard in combination with eCl@ss.

On the client side the customer interface is additionally equipped with some optional features. A female embodied character agent (ECA) named COSIMA embodies the electronic sales agent. Thereby, the animations of Cyberella[30] of the PRESENCE project ([AKGAR99]) are used and combined into an ad hoc created friendly and convincing appearance of the sales agent. For usage of life-like characters in web applications see [AMR97, AR00]. COSIMA does a very emotional job when presenting the search results or bargaining with the customer. For integration of personality ([Dig90]) and emotions into embodied agents see [AKGAR99, Rou96]. Moreover, COSIMA talks to the customer via speech synthesis in real time. Text templates ([RD97]) are used for the text generation to realize speech output e.g. for different sales strategies and to dynamically integrate the discussed quality knowledge about the search results. With the agent based FIPA-OS platform a further high level facility for communication is integrated. With that e.g. technologies for an improved human-computer interaction like speech or mimics recognition can be used. For first experiences with COSIMA^{B2B} in such a visionary scenario see [FDWK03]. Thereby, the emotion modeling is done according to Ekman's six basic emotions ([Ekm82]). The communication for speech input as well as for speech output is done via the standard language VoiceXML ([MBDFH02, BWH02]). The speech input, e.g. to query the database can be done via microphone or keyboard input via natural language. Like in the 'Media Equation' ([RN96]) of the communication researchers Reeves and Nass this prototype is an experiment where a human (sales man) is replaced by a personal computer (e-sales agent). Yet, the focus in this work is not about different "personalities" of computers. With the presented sales agent the evidence is given that people react similarly as to a human sales agent in terms of product selection and bargaining.

Upon the start of COSIMA^{B2B} the friendly embodied character agent welcomes the customer. Then the customer iteratively composes the content of his shopping cart by searching the product database. As illustrated in Figure 5.2 for example the customer is searching for a red storage container made of polyethylene with a volume of about three liters and a width of 100 till 150 millimeters. Actually, there is no perfect match for these search preferences in the product database. Thus, *best alternatives are offered*.

[27] http://www.ssi-schaefer.com
[28] http://www.man-roland.com
[29] http://www.walch.de
[30] http://www.dfki.de/cyberella

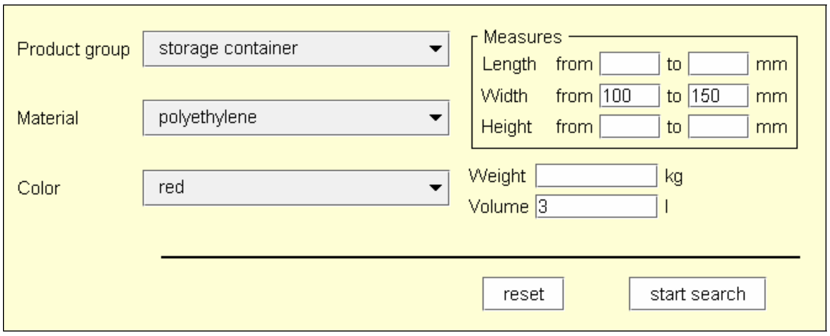


Figure 5.2 Sample customer's search preferences

As shown in Figure 5.3 COSIMA does a smart presentation of the search results. Following a given personalized sales strategy COSIMA *points out a special result* and presents the article with most perfectly fulfilled base preferences, which provides a lot of single sales arguments. In this example she especially emphasizes the perfectly matched red color and fairly mentions the nearly matched volume of 2.7 liters. Because the width is perfectly in the customer's preferred range and also the material is exactly the desired one, COSIMA completes her arguing by emphasizing the 'perfect' overall quality. Finally, COSIMA proactively draws the customer's attention to optional accessories (*cross selling*).

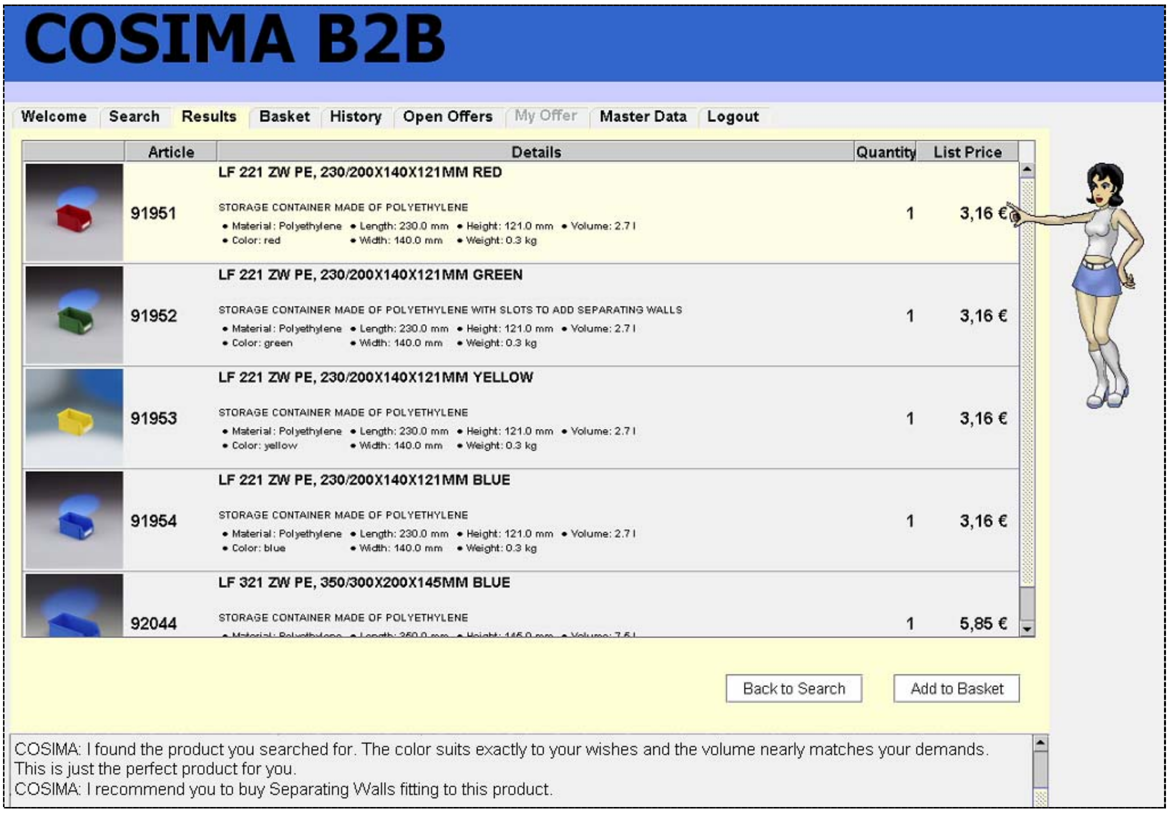


Figure 5.3 Smart search result presentation

After finishing the composition of the shopping cart a personalized and situated price is offered to the customer. Depending on the customer's practice COSIMA offers the opportunity for *further* *price discussions*. During the bargaining process COSIMA makes usage of techniques like *up/cross and down selling*, regarding the customer's and vendor's preferences as well as the situational context (see Figure 5.4).

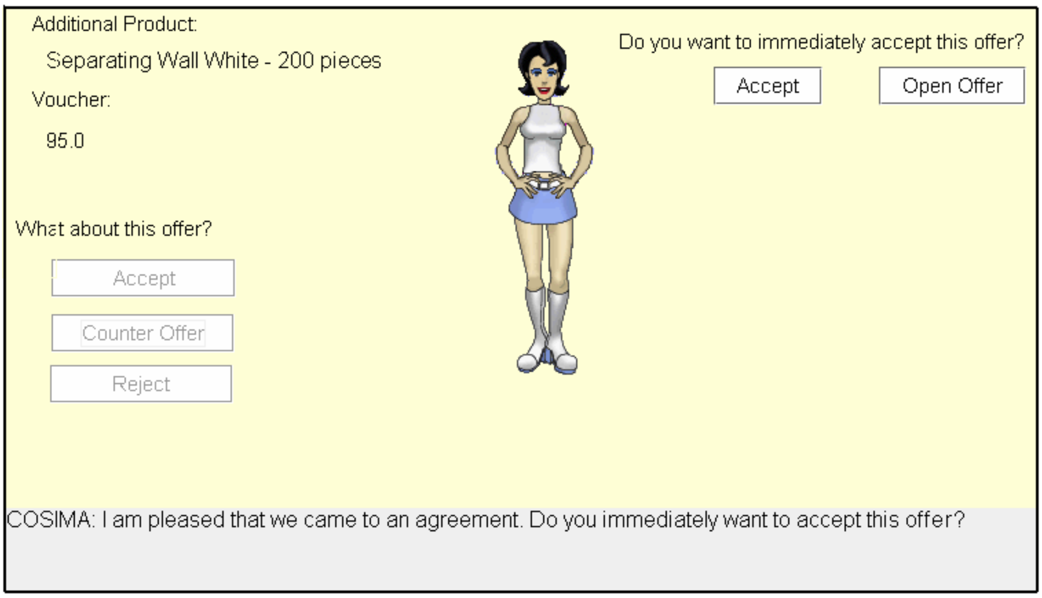


Figure 5.4 End of a successful bargaining session

Besides this, of course COSIMA also provides services to manage the placed orders and open offers as well as the possibility for the customer to give feedback about reasons for a failed open offer.

5.3 The Personalization Manager

Like for a smart human vendor, the behavior of the electronic sales agent is driven by quite a lot of personalized and situated parameters. If the sales strategy is changed by marketing and sales management, then human vendors must be informed and trained to adapt their selling style to this new situation, which can be quite costly and time-consuming. A similar process is necessary for an electronic sales agent, but with the difference that it can be achieved faster. To this purpose a sophisticated sales management tool called the *Personalization Manager* was developed, offering an easy and intuitive interface to adapt the various parameters that drive the whole sales process. The functionality is briefly described, listed as follows:

• *Management of customer master data*: For each customer the master data like name, gender, address, role, etc. can be managed. Name and gender are for example necessary for a personalized welcome. The role is a very important factor influencing e.g. whether a customer is given the opportunity to bargain or not. These data normally are stored in a CRM system and can easily be imported to the personalization manager via xCIL ([Kum02]), and of course also vice versa if master data are updated in the personalization manager.

• *Illustration of customer feedback*: Each feedback for a definitively rejected offer is analyzed by the personalization manager and graphically illustrated individually for each customer. Thus, it is possible to react to deficiencies of the own company, e.g. if the offers are too expensive, the products do not fit well enough the customer's wishes, or competitors offer better general conditions.

• *Management of situated long-term preferences*: For each customer the situated long-term preferences stored in the Preference Repository can be managed manually. Algorithms for Preference Mining like [HEK03, Hol03] can be integrated to automate this potentially expensive process.

• *Management of product search and presentation parameters*: As depicted in the screenshot in Figure 5.5, the importance of customer preferences gained from the search mask can be adjusted. As shown, the material is set to be more important than the color which, in turn, is more important than all the equally important other features. Via various parameters the valuation of the quality of the search results can be adjusted for each customer, e.g. a deviation of up to 10% from the originally required volume should be regarded as 'very good'. As a 'but-only' filter (see section 3.5) the quality claims for each customer are individually respected. E.g. adjustments for the customer in Figure 5.5 demand a high result quality. Even the most promising sales strategy as decided by marketing and sales management can be selected from a pull-down menu. In this example the popular 'second highest price' strategy has been chosen. Using this strategy ensures that a customer is not embarrassed by admitting the price is too high while it shows respect to the customer's financial strength.

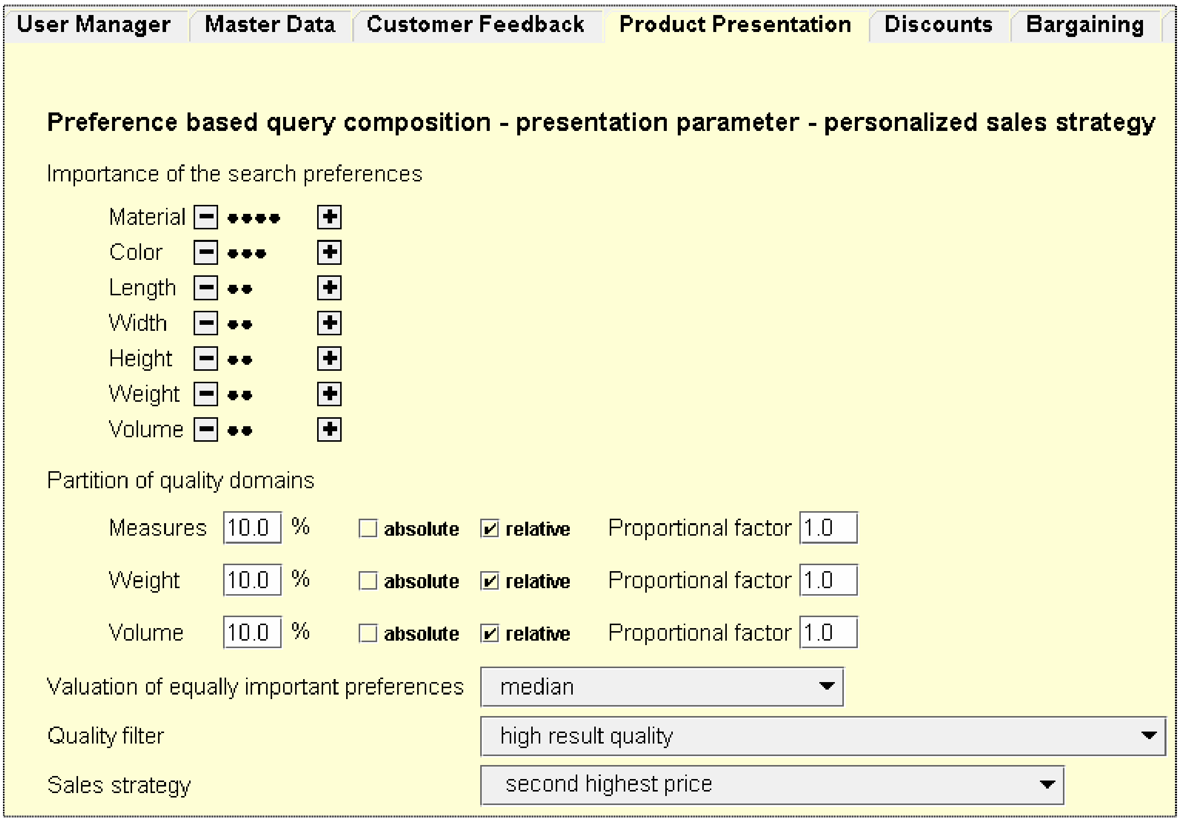


Figure 5.5 Product search and presentation settings

• *Management of price policies*: To adapt to changing price policies, discounts can be adjusted on a personalized and situated basis, enabling the fully automated price fixing. Thereby flexible conditions can be specified, when and how to apply e.g. relative or absolute discounts. If the according discounts are stored in a CRM system, naturally the data interchange can be done via xCIL.

• *Management of the bargaining policies*: Parameters like the probability of up/cross and down selling can be personalized for each customer. From a pull-down menu the overall bargaining strategy can be selected. These parameters can be adjusted individually for each customer. Moreover if the overall sales situation requires this, they can be set globally (see Figure 5.6) to apply to all customers, e.g. to give more vouchers at Christmas.

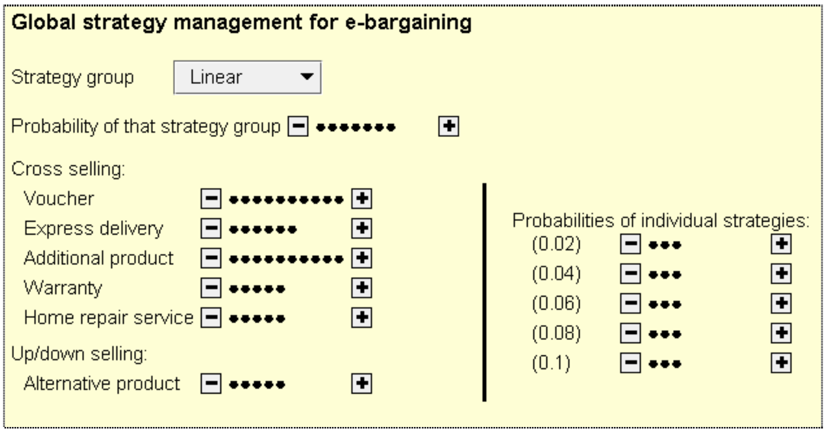


Figure 5.6 Global bargaining adjustments

All this personalization information, except the data from the CRM system, is persistently managed by the Preference Repository.

5.4 Evaluation

In this section a technical evaluation is given to show evidence for the application in a real world e-procurement system. There, the focus is on the search and especially on the presentation part. Afterwards the very interesting and promising results of several evaluations run with test persons on COSIMA^{B2B} are presented, i.e. the impact of the Preference Presenter on human test customers and behaviors of such customers during a bargaining session. For first impressions of a sociological analysis of COSIMA^{B2B} see [FDWK03].

The performance evaluation was done on an Intel Pentium 4, 2.4 GHz and 512 MB RAM computer running under Windows XP. Using Preference XPath real product data, which was provided by the above mentioned industrial partner, was queried. These XML based data of 1.5 MB size included about 1000 products with the full product specifications in BMEcat, including the full feature descriptions in eCl@ss. Almost independent of the number of search results the Preference XPath search averages a little less than two seconds (see Figure 5.7). Naturally, the effort for the presentation calculations rises with the number of results. But only little more than two seconds on the average are necessary for a preference search and presentation. A more detailed performance analysis for the presentation component is shown in Figure 5.8. The quality valuation is done in a couple of milliseconds, depending on the number of results. The quality filter needs about the same effort as the application of the presentation preferences.

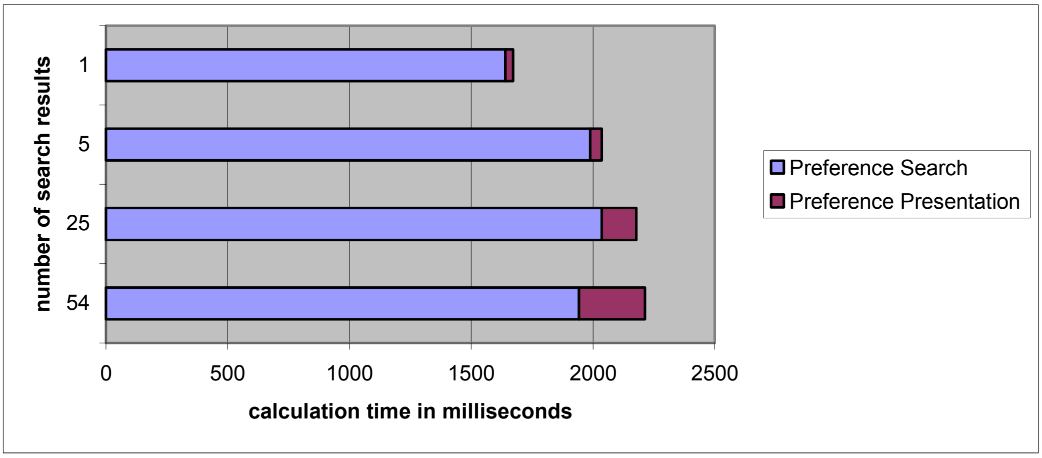


Figure 5.7 Time effort of preference search and presentation

The computation and response time for the calculation of a personalized price offer as well as for an agent's side bargaining bid is about 0.6 seconds ([FKHF02]) and therefore does not reduce the quality of service for an internet customer. Converting and loading speech and animations for the embodied character agent needs clearly less than one second. Indeed almost all test customers in all experimental settings liked the character COSIMA and the speech synthesis.

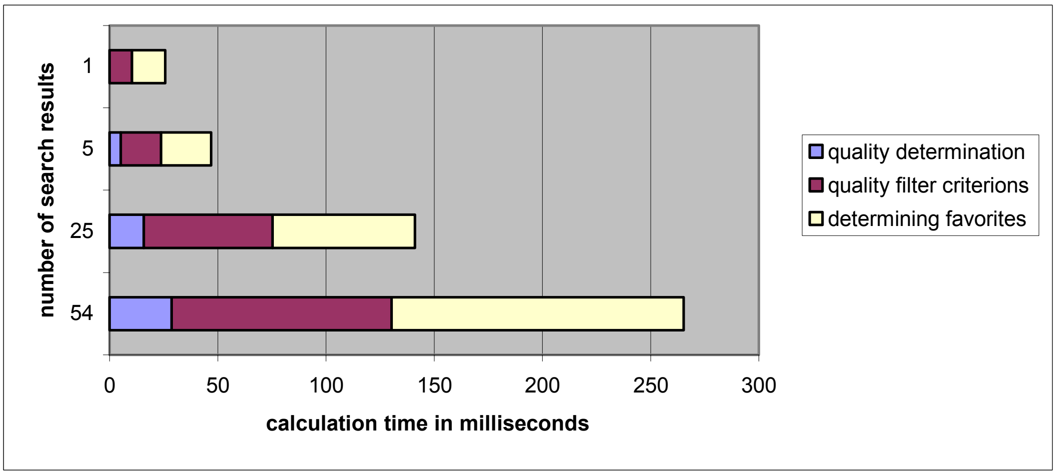


Figure 5.8 Time effort of the Preference Presenter

In an experimental setting 30 test customers had to imagine being in the role of a buyer of an industrial company. Without any special restrictions they had five tasks, i.e. to buy five exactly specified boxes. If there was no such box available they had to select a best alternative on their own responsibility. In average the test customers took nearly half an hour and did their decisions very seriously and carefully. For the quality valuations for all customers the same default adjustments were used. All but two people deemed the quality valuations correct. Thus, a personalization of these parameters is necessary, but obviously an approach of user modeling, e.g. of preference patterns ([BGWK03]) would be appropriate for most of the customers. The customers were analyzed in two groups of 15 people each, by applying a different sales strategy to each group. In the first group the strategy "best overall quality before most perfect arguments" (strategy b) was applied. For the second group the strategy “second highest price before most perfect arguments” (strategy s) was in use. Please note that the result set of course is exactly the same for both groups, but the presentation order can differ.

Querying the properties of the first task, seven boxes in exact the same order for both strategies were delivered by COSIMA. She always recommends of course the first result. As shown in Figure 5.3 nearly 5 results are visible at first sight. It is remarkable that the very seriously deciding test customers briefly scrolled to lower-listed products, but they did not consider their properties. As illustrated in Figure 5.9 only results visible at first sight were selected. Two more people followed the “higher product quality has a higher price” argumentation of strategy s (product 1 is more expensive than 2, 3, and 5) by selecting the recommended result in contrast to people with strategy b. This result is also emphasized by the results of the non-illustrated tasks 2, 3, and 5.

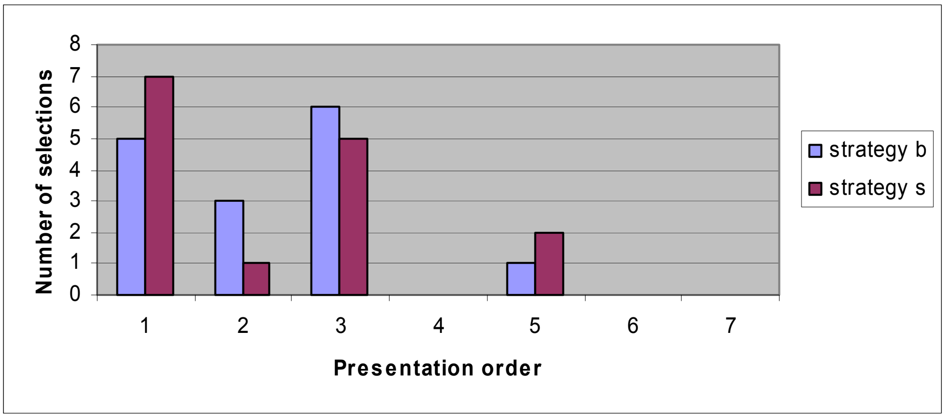


Figure 5.9 Selections regarding the presentation order of task 1

Exemplarily, at last for this experimental setting the experiences of task 4 are illustrated in Figure 5.10. Here different sales strategies followed a different order of the resulting products named V, W, X, and Y. The arguments about the quality of the search result with respect to the search preferences (strategy b) leads to selections of the first ordered products, whereas the “higher product quality has a higher price” argumentation (strategy s) leads to more selections of the product W. Here product Y is the most expensive one.

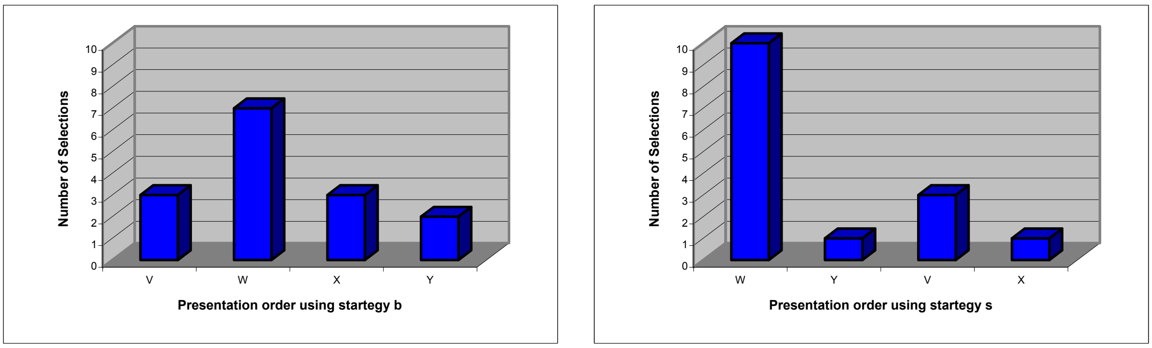


Figure 5.10 Selections regarding the presentation orders of task 4

Summarizing, it can be stated that different sales strategies had indeed different impacts on the test customers. Thereby, not each customer liked each argumentation, e.g. some liked to hear only positive arguments, others expected to hear at least one not perfectly fulfilled search preference in the argumentation. Thus, a *personalization is definitively necessary*. If there is no knowledge about the customer, careful strategies like the 'second highest price' strategy should be applied. As shown in Figure 5.9 this is a *powerful instrument to direct the customer choice behavior*. Moreover, a *good argumentation was experienced as believable* and the test *customers showed a satisfied attitude* during their shopping which is very important for a *successful customer relationship* ([Nic66, EBK78, HS69, Han72]). It is also remarkable that no test customer used the filter button within the preselection. This emphasizes *that the Preference Presenter in combination with a BMO search works very efficiently* and does not bother the customer.

Evaluating the Preference Bargainer component, in a further experimental setting the test customers had to buy a basket of several loosely specified items. Within their decisions they had lots of freedom, but a limited budget. As shown in Figure 5.11 a test customer typically needs about five iterations during the bargaining process, but there are also a few – often unsuccessful – bargains with only one or two iterations.

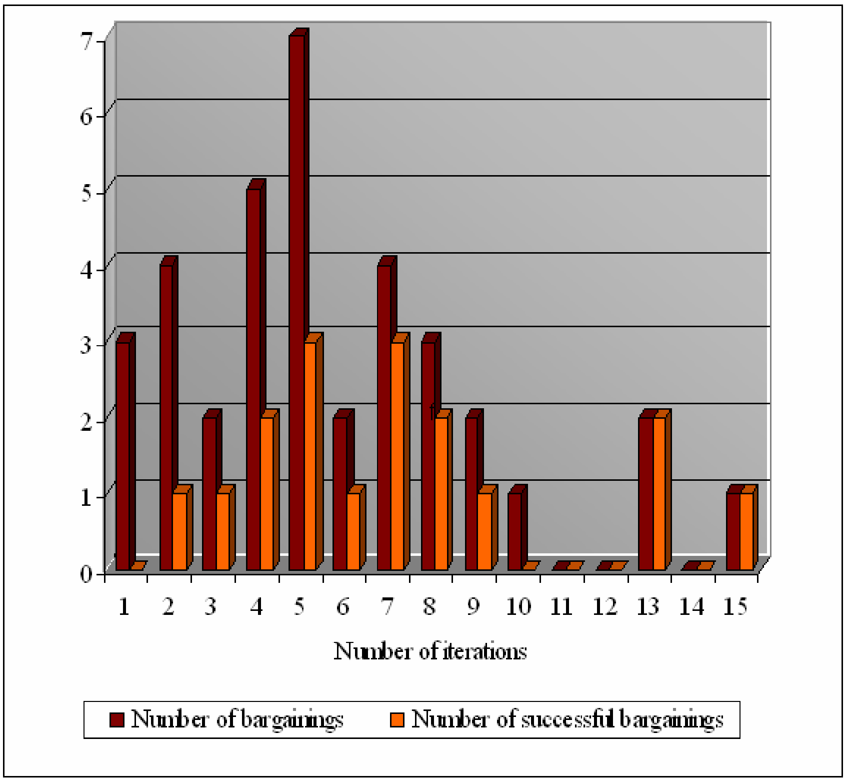


Figure 5.11 Number of bargaining iterations

In these cases the customer does not seem to be interested in the selected product and therefore cancels the bargaining process. Another noticeable observation is that long bargains – 8 - 15 iterations – often lead to a successful deal. In these situations the customer seems to be very interested in the product. He tries to minimize the price or wants some additional services, and tries to achieve his goals during the bargaining process.

In Figure 5.12 the averaged savings of several customers are illustrated. There is an individual starting price for each customer and also a personalized lower limit. The difference denotes the savings potential. The interesting result is that the number of bargaining iterations is independent of the gained savings. This means that a customer is not necessarily kept in the bargaining process. If he deals in a good and fair manner then the customer soon reaches a good deal, which meets the intension.

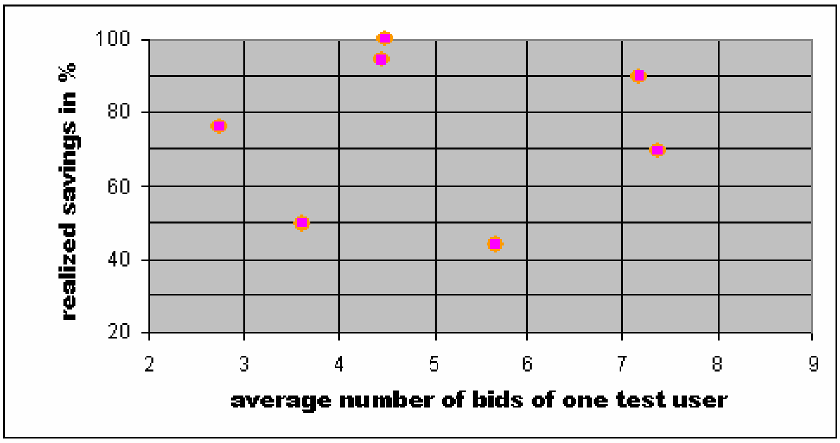


Figure 5.12 Savings gained via bargaining

6 Related Work

The contributions of this thesis are faced to existing approaches. The first section presents the advantages of the novel preference based personalized presentation framework of chapter 3 over previous technologies and approaches, separated into different research areas. The second section exhibits the improvements of this work in terms of the price fixing process.

6.1 Query result presentation

Separated into the following five areas, the advantages and improvements of the preference based personalized presentation framework of chapter 3 are presented. All approaches of the following areas are considered under the same focus, i.e. how to find suitable alternatives to a customer's search request and how to present the alternatives. There are different ways to guide the user to the one product that he finally picks out of a set of alternatives. So, the crucial points in these comparisons are the following:

- Are there approaches that are already able to give human comprehensible reasons for their presented results?
- If not, are there approaches of intelligent database querying which can be modified to deliver good, semantics based arguments about the result quality?
- Perhaps, are there approaches that compute the unique best alternative at once, i.e. no reasoning would be necessary?
- How long does a user need to find the best match; is the user convinced that he found the best alternative?
- Which approaches are able to proactively offer a product, following a given presentation strategy; how flexible is the administration?

In combination with a preference search engine the Preference Presenter does not bother the user with lots of search and presentation iterations. When knowing about the search preferences and the quality sense of the user/customer in a specific situation, as discussed before the Preference Presenter can satisfy these points very well. Even the realization of presentation strategies can be done in a flexible and declarative manner. The necessary knowledge about the search preferences can often be obtained directly out of a search mask or can be gained via Preference Mining techniques ([Hol03, HEK03]), managed in a Preference Repository.

There are already some ad hoc approaches on the web with at least a more human comprehensible overall quality valuation by using a five category model for the quality of a search result as shown in Figure 1.2 or at the German travel agency TravelChannel.de[31]. They valuate a preferred destination with one up to five smilies (see Figure 6.1), but do not provide arguments regarding single characteristics, e.g. the airport of departure.

[31] http://www.TravelChannel.de

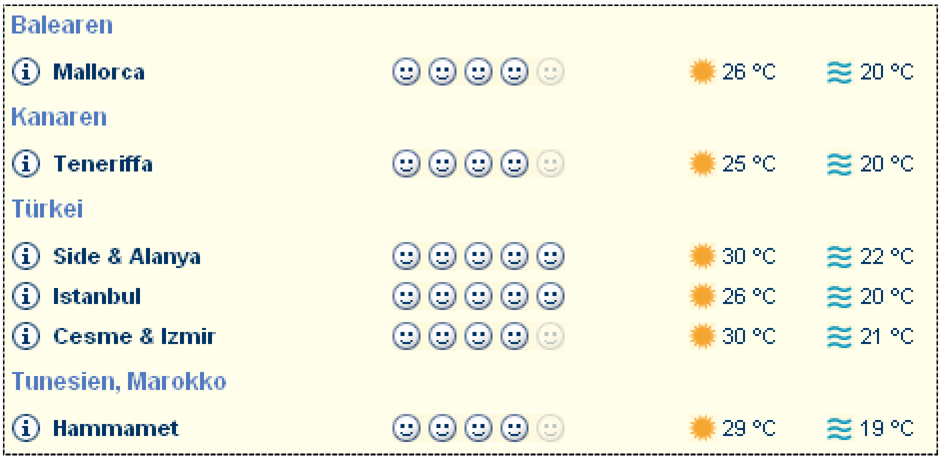


Figure 6.1 Smily valuation of search results

The concerned approaches of the following four reconsidered research areas, i.e. parametric search, fuzzy logic, expert systems, and case based reasoning, have some similarities. They all determine alternative search results via scoring function of about the following form with $\sum w_i = 1$ and for all $i = 1, \ldots, n$: $0 \le dist_i \le 1$:

$$score := \sum_{i=1}^{n} w_i (1 - dist_i)$$

There, $dist_i$ denotes the relative distance regarding a desired characteristic i (i.e. a preference) to the characteristic of the considered alternative. In contrast to the preference model of Kießling ([Kie02, Kie04]) the expressiveness is limited. Equally important preferences as well as more important preferences cannot be combined in general only via scoring functions as shown in [Kie04, Cho03]. As seen in section 3.6 without these expressions (i.e. Pareto and prioritized) several important presentation preferences are prevented. Also providing ranked preferences, moreover, Kießling ([Kie02, Kie04]) provides several intuitive base preference constructors and allows the formulation of preferences in a declarative manner. By basing the valuations of a search result on these intuitive expressions a lot of human comprehensible presentation arguments regarding base preferences as well as complex preferences (ranked, Pareto, and prioritized) are elaborated in this thesis. A preference model limited to a score function of course is also very limited in the variety of presentation arguments, which are necessary for a good reasoning to the user/customer.

6.1.1 Parametric search

In the world of parametric search each database attribute is considered as one parameter, which for customers is often illustrated in so called parametric data sheet tables. This organization of data can be queried very efficiently due to system performance aspects ([WCP02]). Parametric search also considers the case that there might be no perfect match. But instead of computing a set of (best) alternatives it starts with the complete data and the user must iteratively specify conditions. E.g. as shown in Figure 6.2 at Infineon.com[32] there are 14 parameters to adjust. One may specify ranges instead of expressing a preferred value. The intention is to guide the user to his best alternative product from the very beginning of the search. Then a quality valuation is not necessary. But, of

[32] www.infineon.com

course, this is a very time-consuming and frustrating process. A sample result is shown in Figure 6.3. After several refinements of the parametric search parameters still 72 matches are left.

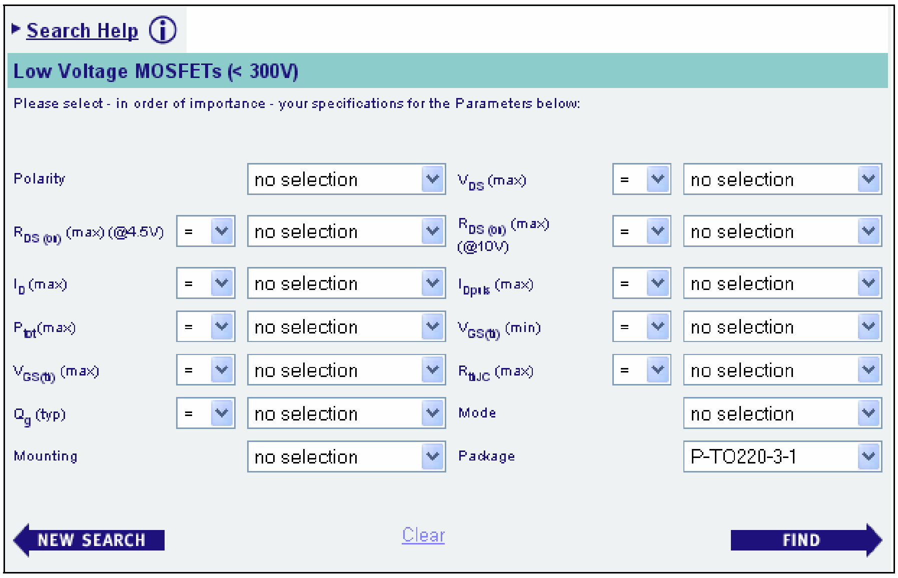
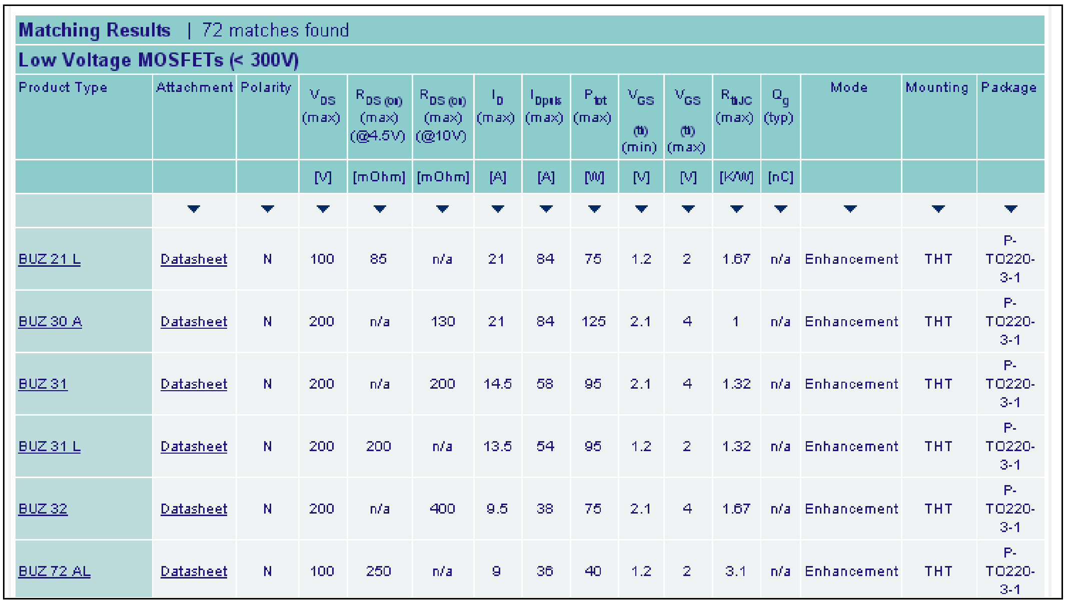


Figure 6.2 Parametric search adjustments at Infineon

The problem here is that a human customer needs a deterministic algorithm to find the best alternative in this multi-decision process. Adjusting one parameter may destroy the improvements obtained by the last adjustments. Moreover, it often leads to the empty result effect or to the flooding effect with irrelevant results. This is unreasonable for the user. Therefore, to fight the flooding effect, parametric search is suggested to be used with a ranked preference model by Agrawal and Wimmer ([AW00]).


Matching Results | 72 matches found

Low Voltage MOSFETs (< 300V)

Product Type	Attachment	Polarity	V_{DS} (max) [V]	$R_{DS(on)}$ (max) (@4.5V) [mOhm]	$R_{DS(on)}$ (max) (@10V) [mOhm]	I_D (max) [A]	I_{Dpuls} (max) [A]	P_{tot} (max) [W]	$V_{GS(th)}$ (min) [V]	$V_{GS(th)}$ (max) [V]	R_{thJC} (max) [K/W]	Q_g (typ) [nC]	Mode	Mounting	Package
BUZ 21 L	Datasheet	N	100	85	n/a	21	84	75	1.2	2	1.67	n/a	Enhancement	THT	P-TO220-3-1
BUZ 30 A	Datasheet	N	200	n/a	130	21	84	125	2.1	4	1	n/a	Enhancement	THT	P-TO220-3-1
BUZ 31	Datasheet	N	200	n/a	200	14.5	58	95	2.1	4	1.32	n/a	Enhancement	THT	P-TO220-3-1
BUZ 31 L	Datasheet	N	200	200	n/a	13.5	54	95	1.2	2	1.32	n/a	Enhancement	THT	P-TO220-3-1
BUZ 32	Datasheet	N	200	n/a	400	9.5	38	75	2.1	4	1.67	n/a	Enhancement	THT	P-TO220-3-1
BUZ 72 AL	Datasheet	N	100	250	n/a	9	36	40	1.2	2	3.1	n/a	Enhancement	THT	P-TO220-3-1

Figure 6.3 Sample result of Infineon's parametric search

This can efficiently be answered regarding system performance as shown in [HKP01, HP04]. Agrawal and Wimmer ask customers to specify weights of importance between 0 and 1 for each attribute/parameter specified in the parametric data sheet table. This again is a very time intensive and bothering task. A simple score is calculated and the ‘top-k’ scored results are presented ([HKP01]), ‘top-k’ in terms of that score function.

The improvement for the customer supposedly lies in the automatic computation of this otherwise iterative adjustment process. But there occurs the well known problem of such a scoring function for a customer. He can still be not sure whether there is a better fitting alternative, because a human being does not think in mathematical distances of weighted metrics. Moreover, if the scores are too bad or zero, then users would even be asked to soften or skip some of the search conditions by adjusting them in the parametric data sheet table, once again.

One problem of this approach is the very high and often repeatedly necessary human interaction, which is a very time-consuming process. This might be improved by Agrawal’s and Wimmer’s approach, but it still takes a very long time when adjusting the weights. In this approach, where the computer automatically searches for alternatives for the user, there is once again the question about the result quality. Of course with the numerical preference and the valuation of the numerical preference as described in Definition 2.9/Definition 3.28 the search can also be expressed and the search results can even be valuated with the achievements of this thesis. But when expressing the search criteria with only one simple score, the quality characteristics regarding base preferences are blocked and therefore also valuable presentation arguments. Hence, several presentation preferences can not be covered with parametric search.

6.1.2 Fuzzy logic

Fuzzy logic is a form of mathematical logic in which truth can assume a continuum of values between 0 and 1 ([Zad73]). No known work in the area of fuzzy logic provides a comprehensible quality valuation and presentation of search results. Yet, there are some fuzzy logic approaches in the areas of recommender systems and decision support ([Yag03, KIW02]) that perhaps can be adapted for presentation. Yager’s ([Yag03]) fuzzy approach, to decide which result to present first aims to find a *mathematical similarity between objects*, i.e. here between the desired perfect match to be recommended according to the user’s preferences and a possible alternative. A user’s preferences are modeled by recording that he likes or dislikes something referring a basic characteristic ([Yag03]). E.g., one can express he likes blue cars, but that does not mean he prefers blue over other colors. In contrast to the preference model of [Kie02, Kie04] no alternatives are mentioned in this model. Besides very domain dependent rule based approaches, the procedure of fuzzy approaches is mainly about the following. The preferences p_i are either 0 or 1, i.e. satisfied or not. The similarity measurement of an object ob and the perfect object, which gives the degree of satisfying a complex preference, is gained by summed weighted base characteristics with weights w_i ($\sum w_i = 1$) in the form:

$$\text{similarity}_{\text{perfect\_object}}\,(\text{ob}) := \sum_{i=1}^{n} w_i p_i$$

The preference weights w_i must be known for each user in order to calculate this complex preference. Up to that point, the above approach of Agrawal and Wimmer is very similar. In the fuzzy logic there are also lots of variations of similarity measurements (e.g. [Yag03]), i.e. how to combine the single characteristics into a complex preference statement. Yet, in the end there is always a number between 0 and 1. The founder of fuzzy logic theory, Zadeh, realized that people do not understand numbers in many domains. Therefore as described in section 3.2 he introduced human comprehensible linguistic variables. Zadeh also introduced linguistic selection criteria regarding characteristics p_i like 'at least', 'at least one', 'some', 'all', and 'at least α%' ([Zad83]). In the recommender system of [Yag03] the 'at least α%' (see Figure 6.4) quantifier denotes a threshold of the similarity key data, whether a result should be recommended or not. This could be used as a quality filter criterion.

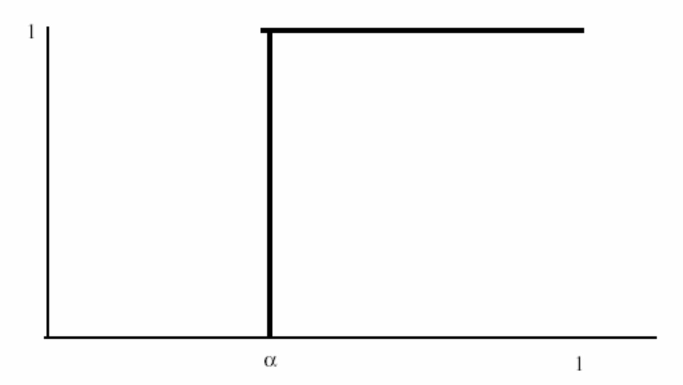


Figure 6.4 Linguistic quantifier "at least α %"

Sedbrook ([Sed98]) works with some single preferences about the weather e.g. "he likes sunshine". He uses several thresholds for his linguistic model about the preferred weather (Figure 6.5). Combining e.g. Yager's one-dimensional 'at least α%'-criterion with Sedbrook's six an -at first sight human comprehensible- overall quality could be calculated.

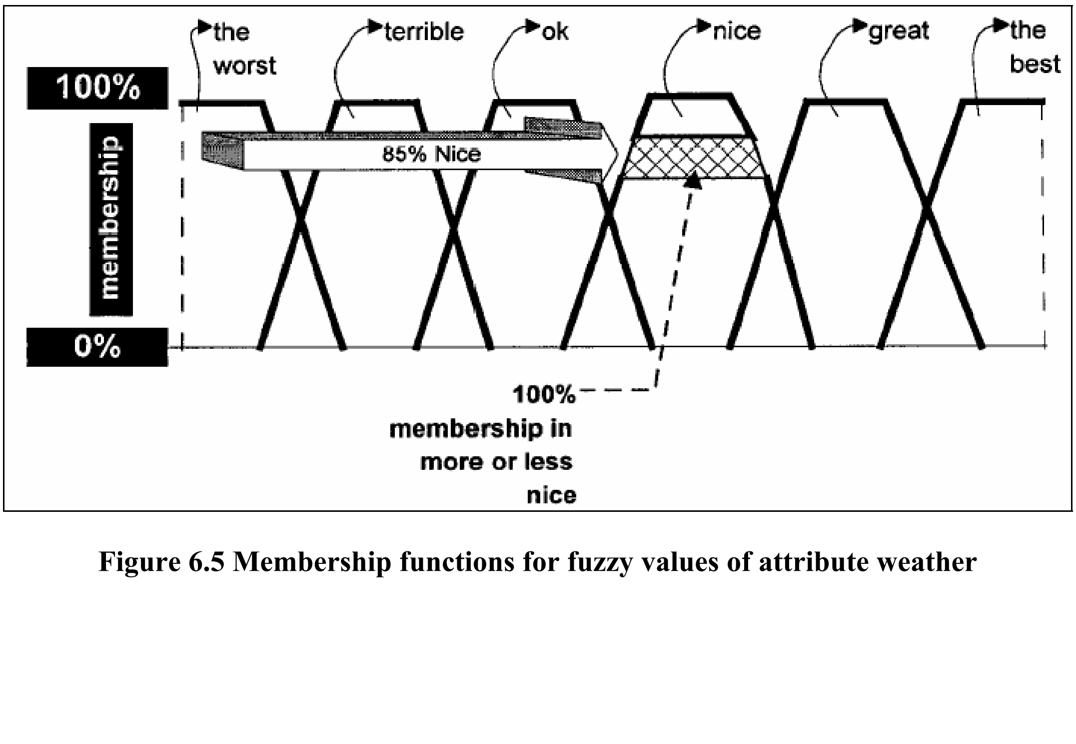


Figure 6.5 Membership functions for fuzzy values of attribute weather

If the weights w_i of user preferences do already exist in a user profile ([Yag03]) then the user is not bothered with lots of search iterations, provided that he is happy with an ordered list according to the similarity measure wrapped into a linguistic variable. Otherwise, the customer must specify the weights, which is very time-consuming. Please note that this approach would also be realizable by search and valuation of a numerical complex preference as described in section 3.4.3.

Because of the restricted underlying preference model ("I like A" – "I dislike B") of the described approaches no qualities other than true or false can be obtained for single result characteristics. Moreover, there are conflicts in the modeling of single characteristics, e.g. one may have a preference POS(color, {blue, red}). In the discussed fuzzy approach this would be formulated with two preferences, namely "I like blue" and "I like red". But even when the color preference of this customer is satisfied, one preference of this restricted model is not fulfilled. The Preference Presenter is based on a semantically much more powerful and declarative preference model ([Kie02, Kie04]) that enables the modeling of much richer arguments as seen before. Hence, the discussed fuzzy logic approaches also prevent several discussed presentation preferences.

6.1.3 Expert systems / knowledge based systems

Expert systems, also known as knowledge-based systems, are software agents designed to simulate the problem-solving behavior of human experts within very narrow domains by asking the human several questions. One example is the multiple attribute decision making system MADM ([MZFLZ01]). In MADM the expert systems need preference information on alternatives, i.e. the alternatives must already be known to an object. The goal of MADM is to rank the alternatives or to select the most desirable one. In contrast to MADM, the Preference Presenter never orders all results, but according to a presentation strategy marks one or more most desirable ones. Disregarding the formalism of MADM the proceeding is, like for all of the known knowledge based approaches, a fuzzy approach. The simulated experts model alternative preference information as fuzzy relations and use that in a probabilistic combination to rank the results. The result with the best score is the one to be pointed out.

Obviously, a first problem occurs with gaining preference information for the experts. A similar and concrete approach is given by Burke with a knowledge-based restaurant recommender system ([Bur99]). As discussed before a recommender can possibly be reengineered to enable a comprehensible result presentation as described above. Yet, also Burke bothers a user several times, asking him to refine the search preferences, e.g. to broaden the price range for a meal etc. Then he calculates the then possible best alternatives with ranked similarity measures.

As described above mathematical similarity measurements are limited expressions to formulate preferences. Moreover, expert systems are very domain dependent. There is no declarative manner for instantiating the expert system for the search in a new domain such as the one available for the preference search and the presentation preferences. In contrast to knowledge based approaches the Preference Presenter does not need several time-consuming iterations of customer interaction, because it delivers best matches according to the BMO model. No iterative refinement of search conditions is necessary. Of course, with the numerical preference the preference information of the expert system can also be used in a BMO search. Furthermore, the Preference Presenter is able to provide semantics based arguments to support a suitable reasoning, because it is based on a semantically rich preference model. Presentation strategies, even with conflicting utility functions, can easily be formulated in a flexible, semantic manner.

6.1.4 Case based reasoning

Case based reasoning is a technique for problem solving that looks for previous examples which are similar to the current problem. There are many situations where experts are not happy to be questioned about their knowledge by people who want to capture the knowledge in rules for use in expert systems. In most of these situations, the natural way for an expert to describe his knowledge is through examples, stories, or cases, which are all basically the same thing. Such an expert teaches trainees about the expertise by apprenticeship, i.e. by giving examples and by asking the trainees to remember them, copy them, and adapt them in solving new problems if they describe situations that are similar to the new ones. Case based reasoning aims to exploit such knowledge. In contrast to expert systems and fuzzy logic, case based reasoning *considers similarities between people,* not between objects. Case based reasoning is often used to realize a *collaborative filtering* approach, which aims at *exploiting preference behavior* and qualities of other persons in speculating about the preferences of a particular individual. This technique perhaps might help to find the unique best alternative automatically.

There are lots of approaches in e-commerce to assist the customer with the search engine, e.g. in [MA01, RAMV02, XAF03, JSZC03]. They all compute the similarities between the customers in a different way, in some cases combined with previously described approaches. Yet, what all have in common is that they introduce a score in form of a mathematical similarity measure that determines how similar the buying behavior of different users is. With that information they e.g. recommend the same products or start with the same search adjustments. If a customer is not satisfied with any recommended result, the PCFinder system of [XAF03] automatically adjusts some preference weights and tries again and again. The customer never experiences whether the offered result is really the best alternative for him. He cannot know when to stop this iteration. [JSZC03] also tries to predict the preference behavior of a customer by adjusting weights.

Because the described similarity measures for an automatic alternative determination often fail there are also more manual approaches. But perhaps this could be a more comprehensible way for a customer as realized e.g. by [RAMV02]. The ITR travel advisory system acts in the frequent case that people do not find an exact match for their preferred holiday trip. If this empty result effect occurs, ITR helps with case based reasoning to relax the search constraints. Similar cases are once again calculated by a similarity measure. Like in a one-dimensional parametric search the customer is now asked to soften or to skip a search constraint step by step (see Figure 6.6). Yet, that of course implies a lot of customer interaction.

Figure 6.6 Query relaxation guided by ITR

Of course, when knowing the preference behavior this ranked approach can also be modeled with the introduced preference search and presentation via numerical preference. The problems once again lie in limited expressiveness, thus in the limited presentation arguments, and also the limitations regarding the presentation strategies. Because of the efficient filter effect of a BMO search ([KFHE01]), also the user/customer interaction can be reduced enormously by presenting all relevant best matches at once.

6.1.5 E-catalogs

E-catalog providers, too, have recognized that the presentation order of the results is very important. For that purpose in e-catalog systems like ems-media's PowerCatalog it is possible to mark favorites in the underlying product database. Query results whose favorite flag is true are ordered first, then the rest. A similar simple technique is provided by electronic catalog structures of Poet. Special manufacturers can be specified, to be placed higher in the presented result set than others, e.g. if there is a product from IBM, put it on top. This is a very pragmatic way. In contrast to a declarative approach like the Preference Presenter there is a lot of work to do to administer and update the data in this inflexible and static flag approach. Obviously, with a preference search these simple presentation strategies can be formulated with a POS(favorite_flag, {true}) and POS(manufacturer, {IBM}), respectively. Please note, for the manufacturer preference the database need not to be manipulated. The rule can be declaratively formulated in the moment of the search and can flexibly be managed in the Preference Repository, personalized for each customer. Furthermore in an e-catalog, there are no arguments to do a good reasoning facing the customer. Also, presentation strategies are hard to realize. A simple sales strategy like ‘second highest price’ is not combinable with the hard-coded favorites, which always appear first followed only then by the rest. This conflict cannot be handled by this approach.

Summarizing, there is no approach which automatically can compute the only one fitting alternative for a customer wish in case of a missing perfect match. Because of the underlying semantic rich and expressiveness preference model, the Preference Presenter is the only known approach with user/customer comprehensible arguments regarding base and complex preferences for explaining the results and to make clear to the customer why a particular search result is suitable. The just discussed approaches block arguments regarding base preferences by only considering an overall score. They are not able to argue to the customer and often imply a very high customer interaction with the system. The Preference Presenter avoids this problem by only delivering best matches followed by a good reasoning. Furthermore, the Preference Presenter is the most flexible and comprehensible approach for realizing a presentation strategy, amongst others based on search result qualities, by formulating them as preferences in a declarative manner. This enables even the specification of conflicting requirements. Other approaches prevent several important presentation strategies since they have a limited expressiveness in the query model followed by missing presentation arguments.

6.2 Price fixing

Two novel approaches have been introduced in this thesis, a structure for electronic data interchange to enable a personalized price offer and a preference based multi-objective electronic bargainer.

6.2.1 Personalized price offer

In [KLS02] the importance of a comprehensive structure for data interchange for different price/discount models is extensively discussed by means of existing catalog structures like BMEcat. For an e-procurement portal, according to the business data processing specialists Kelkar, Leukel, and Schmitz ([KLS02]) the following three types of pricing are relevant and absolutely necessary:

- ♦ Individual pricing (a personalized price for each product)
- ♦ Quantity pricing (differential prices, separate for each product)
- ♦ Bundled pricing (prices for product bundles, cheaper than the price sum of the single products)

Like proposed in this thesis, the first two types are realizable by BMEcat, while bundled pricing and the very important calculation order is only supported by the not very popular EAN UCC[33] format that misses lots of other characteristics. In this thesis the demands according to [KLS02] are consequently realized with the DTD structure of Appendix B as a first standardization approach. This leads to the resulting personalized price offer technology. With this, all the deficiencies regarding the price discussed in [KLS02] of an e-procurement store are overcome.

[33] http://www.ean-ucc.org

6.2.2 Preference Bargainer

The positive effect of bargaining in e-commerce has been already realized by [LD99]. A first commercial attempt is done by ONE Smart World[34] with signBazar. There automatically an email with the shopping cart and a manually added price bid is sent to the vendor side, which then manually has to calculate a price and send a reply with a counter bid. This is done iteratively until there is an agreement or a final disagreement between customer and vendor. With [LD99] an automated way is shown in which customers bargain with an electronic agent. Yet, this is limited to one single product and the bargaining is only about the price. Moreover, only three bidding strategies are in use. With the introduced Preference Bargainer, a much more powerful technology is given, by allowing the flexible integration of each kind of strategy. The Preference Bargainer uses about 40 strategies at the moment while learning and respecting the preferred strategies, personalized for each customer. Moreover, the bargaining is about a product bundle as demanded for the price computation in [KLS02]. And last but not least, the techniques of up/cross and down selling that are typical for bargaining are supported fully automatically, including the underlying product semantics. So, in contrast to current approaches this is a very advanced framework.

[34] http://www.one-smartworld.de

7 Summary and Outlook

In this final chapter at first this thesis is summarized, especially pointing out the essential achievements. Then, an outlook and suggestions for future work complete this thesis.

7.1 Summary and achievements

After analyzing the continuing misery of today's search engines and the problematic point of search result presentation in general, a further problem especially for e-procurement applications, i.e. a personalized offer composition was discussed. Following that, a review of the preference model of [Kie02, Kie04] and the achievements around this model and its semantic features have been given. Moreover, it has been elaborated that the knowledge about the quality of the search results with respect to the search preferences in a specific situation is a major issue with regard to several aspects, e.g. for consumer choice behavior in e-commerce applications ([Nic66, HS69, EBK78, Han72]). Then in chapter 3 till 5 the following achievements were introduced:

- **Personalized query valuation and presentation of a query result**
  For the crucial process of the search result presentation a novel effective approach has been elaborated.
  - Quality valuation of single search preferences
    Following the philosophy of [Zad73] a linguistic model for the search result quality has been introduced. An intuitive framework has been elaborated for situated quality valuations by satisfying the postulate that worse results must not have a higher valuation than better results. Defining quality valuation functions for all base preference constructors of [Kie02, Kie04] with an output of linguistic quality terms an extensible framework for a *human comprehensible search result presentation* was created, followed by *valuable presentation arguments* regarding each single search preference.
  - Calculation of a situated overall quality
    Based on these results various situated quality valuations for complex preferences ([Kie02, Kie04]) have been introduced, also discussing wrong, incomprehensible valuations by an example. With that, an algorithm has been presented to calculate a personalized and situated overall quality of a search result, including the computation of the quality of each occurring base preference.
  - Presentation strategies
    After respecting a so called "but-only" filter, i.e. filter criteria applied over the search results, various intuitive selection criteria have been introduced for deciding which result to present first or which results to proactively offer to a customer, respectively. Especially for the case of a sales process, selection criteria have been presented. For the first time this enables the flexible application of personalized sales strategies ([Rac89, HS98]) within an e-commerce sales process. This can be done in a declarative manner, even coping with conflicting presentation preferences.

Realized as the Preference Presenter technology in combination with a preference search engine, the search and presentation process can be managed very efficiently for a user/customer. The customer does not have to iterate through several pages and moreover gets helpful arguments why the/these result(s) are the suitable one(s).

- **Personalized offer composition**
  After discussing the deficiencies of state-of-the-art e-procurement technology preference technology was used to build up an automated personalized offer composition process.
  - Product composition
    With the Preference Presenter a major step has been done towards a human comprehensible product presentation. The Preference Presenter, a preference search engine like Preference XPath, and the Preference Repository, have been used to provide an effective and efficient composition of the shopping cart within an e-procurement process. There, current search preferences, long-term customer preferences as well as vendor preferences are considered.
  - Price fixing
    For an efficient online price fixing two novel technologies have been designed. First to realize a personalized price offer an XML based structure for electronic data interchange for complex discounts has been modeled. This easily allows the automatic calculation of a personalized price for a given product bundle. Second with the Preference Bargainer a novel technology has been designed for multi-objective autonomous bargaining, including the usage of the typical bargaining techniques up/down and cross selling.
  - Standardized data integration
    By analyzing current e-procurement standards like BMEcat and eCl@ss an effective way for a standardized integration of price and product data has been modeled. With that, the necessary economical knowledge for an autonomous e-procurement sales agent is provided.
- **Engineering of an e-sales agent**
  The consequent realization of the contributions introduced brings evidence of effectiveness and efficiency.
  - Realization
    With COSIMA^{B2B} ([KFD04]) a deeply personalized and situated prototype of an emotional sales agent has been presented and it has been shown how the novel contributions of this thesis can automate such a time and cost intensive sales process. The personalization manager enables a comfortable adjustment of personalization parameters.
  - Evaluation
    Several experiments with test customers as well as of technical manner showed the applicability of COSIMA^{B2B} regarding technical aspects as well as according to the effect on human beings. Especially, it was shown that sales strategies applied by a computer instead of a human have about the same effect to a human customer.

7.2 Outlook and future work

A next step is to analyze the automation potential of the B2B buyer side. All novel technologies introduced are interoperable, and also a single use of a component is possible. Especially the Preference Presenter is a promising technology to support a human customer with his decision by respecting his selection preferences. Even an automated decision within a buying process via Preference Presenter is imaginable, e.g. for a company-internal BMEcat marketplace like MAN2B[35].

Further fields of applications for the Preference Presenter could be comparison shops like COSIMA ([KFHE01, KHFE01]). In that platform a fixed heuristic that determines the presentation order is implemented equally for all customers. With the Preference Presenter, in such a platform each customer could be easily and flexibly treated according to his search quality claims and with respect to his presentation preferences. Moreover, some strategies of the service provider could be respected, e.g. to flexibly push sponsored results without conflicting with the customer's preferences. Also an electronic trustee as described in [DFGSW04] can benefit from the Preference Presenter technology. Moreover, cooperative web services can be supported within their decision process. E.g. in order to find the best fitting web service for a special task, in [BW03] a soft selection according to the BMO model is suggested. Obviously, the Preference Presenter is predestined for managing a personalized decision support. A final example is a personalized notification service like P-NEWS ([BGWK03, WBKH04]). It is the development of a preference driven news service in MPEG-7 libraries ([Smi00, Smi01]). In this project the Preference Presenter technology has been integrated currently as basis for the illustration of the search quality and for making the decision which documents or objects to notify to a registered customer.

The knowledge for the adjustment of the partitioning parameters is at the moment a rather manual task by the knowledge engineer of the search engine. For an update due to changes, e.g. of the product domain, information agents were suggested in this work. But many other influences can change the quality sensation of a customer and preferences can change with the according situation. This dynamic aspect for the definition of a situation is discussed e.g. by the sociologist Thomas ([Tho23]) or from a different point of view by the human factors researcher Endsley ([End95]). Therefore, for the general detection of partitioning parameters for the quality functions an automated process would be very helpful. An imaginable solution could be a technique like Preference Mining ([Hol03, HEK03]).

A completely orthogonal enhancement of this work is to combine the technologies introduced, especially the COSIMA architecture with further technologies for a better human-computer interaction, which can be improved by means of using emotions ([CDTVK01]). As shown in [FDWK04] via FIPA-OS technologies like mimic recognition can be integrated. This recognition of the customer's current emotion can improve e.g. the bargaining process because of a better understanding of the customer's actions. A next step might be the usage of emotional speech synthesis. For an overview of existing technology see [Sch01]. Also the recognition of emotions via speech input ([MO99]) and the correlation with the detected emotions via mimic in such an e-procurement scenario is an interesting topic.

[35] https://www.man2b.com

Literature

[AHU74] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman: *The design and analysis of computer algorithms*. Addison-Wesley Publishing, 1974.

[AKGAR99] Elisabeth André, Martin Klesen, Patrick Gebhard, Steve Allen, and Thomas Rist: *Integrating Models of Personality and Emotions into Lifelike Characters.* In proceedings of the workshop on Affect in Interactions – Towards a new Generation of Interfaces in conjunction with the 3rd i3 annual conference, Sienna, Italy, 1999, pp. 136-149.

[AMR97] Elisabeth André, Jochen Müller, and Thomas Rist: *WebPersona: A LifeLike Presentation Agent for Educational Applications in the World-Wide Web.* In proceedings of the Workshop on Intelligent Educational Systems on the World Wide Web at AI-ED'97, 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, August 1997, pp. 78-85.

[AR00] Elisabeth André and Thomas Rist: *Adding life-like synthetic characters to the web.* Cooperative Information Agents IV, Lecture Notes of Artificial Intelligence, 2000, pp. 1-13.

[AW00] Rakesh Agrawal and Edward Wimmers: *A Framework for Expressing and Combining Preferences*. In proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, TX, USA, 2000, pp. 297-306.

[Bän85] Axel Bänsch: *Verkaufspsychologie und Verkaufstechnik*. Oldenbourg Verlag, 1985.

[BCG02] Nicolas Bruno, Surajit Chaudhuri, and Luis Gravano. *Top-k Selection Queries over Relational Databases: Mapping Strategies and Performance Evaluation*. ACM Transactions on Database Systems, Volume 27, Issue 2, 2002.

[Bec98] Walter Becker: *Verkaufspsychologie – Theoretische Grundlagen und praktische Anwendungen*. Profil Verlag, 1998.

[BGWK03] Wolf-Tilo Balke, Jose Gonzalez-Pinto, Qiuyue Wang, and Werner Kießling: *P-NEWS: A Personalized Notification Service for MPEG-7 Libraries*. Technical Report 2003-2, University of Augsburg, 2003.

[Bir79] Vera F. Birkenbihl: *Psycho-logisch richtig verhandeln*. Moderne Verlag, 1979.

[BPSM00] Tim Bray, Jean Paoli, Michael Sperberg-McQueen, and Eve Maler: *Extensible Markup Language (XML) 1.0*. http://www.w3.org/TR/REC-xml, 2000.

[Bur99] Robin Burke: *Knowledge-based recommender systems*. Encyclopedia of Library and the AAAI Workshop on Artificial Intelligence for Electronic Commerce, AAAI Press, 1999.

[BW03] Wolf-Tilo Balke and Matthias Wagner: *Cooperative Discovery for User-centered Web Service Provisioning.* In proceedings of the International Conference on Web Services (ICWS '03), Las Vegas, NV, USA, June 2003, pp. 191-197.

[BWH02] Daniel Burnett, Mark Walker, and Andrew Hunt: *Speech Synthesis Markup Language Specification*. http://www.w3.org/TR/speech-synthesis, 2002.

[CD99] James Clark and Steve DeRose: *XML Path Language (XPath)*. http://www.w3.org/TR/1999/REC-xpath-19991116, 1999.

[CDTVK01] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John Taylor: *Emotion recognition in human-computer interaction*. IEEE Signal Processing Magazine, Volume 18, Number 1, January 2001, pp. 32-80.

[Cho02] Jan Chomicki: *Querying with Intrinsic Preferences*. In proceedings of the International Conference of Advances in Database Technology (EDBT), Prague, Czech Republic, March 2002, pp. 34-51.

[Cho03] Jan Chomicki: *Preference formulas in relational queries*. ACM Transactions on Database Systems (TODS), Volume 28, Issue 4, December 2003, pp. 427-466.

[CP03] Stephen Cranefield and Martin Purvis: *Referencing Objects in FIPA SL: An Analysis and Proposal.* In proceedings of the 2nd International Workshop on Challenges in Open Agent Environments at the second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003), Melbourne, Australia, 2003.

[DFGSW04] Claus Dziarstek, Frank Farnschläder, Sandra Gilleßen, Irene Süßmilch-Walther, and Veronica Winkler: *A User-Aware Financial Advisory System*. Multi-Konferenz Wirtschaftsinformatik 2004 (MKWI 2004), Essen, Germany, 2004.

[DFN01] Rajiv Dewan, Marshall Freimer, and Paul Nelson: *Impact of search engine ownership on underlying market for goods and services*. Thirty-Fourth Annual Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii, January 2001.

[Dig90] John M. Digman: *Personality structure: Emergence of the five-factor model*. In M. R. Rosenzweig & L. W. Porter (Eds.), Annual review of psychology: Volume 41, Palo Alto, CA, USA, 1990, pp. 417-440.

[Dom02] Jim Domanski: *Add-On Selling: How to Squeeze Every Last Ounce of Sales Potential From Your Calls*. Business by Phone, 2002.

[DP90] Brian Davey and Hilary Priestley: *Introduction to Lattices and Order*. Cambridge Mathematical Textbooks, Cambridge University press, 1990.

[Dut96] Thierry Dutoit: *High-Quality Text-to-Speech Synthesis: an Overview*. Journal of Electrical & Electronics Engineering: Special Issue on Speech Recognition and Synthesis, Volume 17, Issue 1, Australia, 1996, pp. 25-37.

[EBK78] James F. Engel, Roger D. Blackwell, and David T. Kollat: *Consumer Behavior*. Dryden Press, USA, 1978.

[Ekm82] Paul Ekman: *Emotion in the human face*. Cambridge University Press, New York, 1982.

[End95] Mica R. Endsley: *Toward a theory of situation awareness in dynamic systems*. Human Factors. 37.1, 1995, pp. 32-64.

[EZ02] Werner Esswein and Sabine Zumpe: *Realisierung des Datenaustauschs im elektronischen Handel*. Informatik-Spektrum, 8/2002, Springer-Verlag, August 2002, pp. 251-261.

[FDWK04] Stefan Fischer, Sven Döring, Matthias Wimmer, and Antonia Krummheuer: *Experiences with an Emotional Sales Agent*. Lecture Notes of Computer Science, Volume 3068/2004: Affective Dialogue Systems: Tutorial and Research Workshop (ADS 2004), Kloster Irsee, Germany, June 2004, pp 309-312.

[Fis87] Gert Heinz Fischer: *Praxis der Interaktionsstrategie im Verkauf und Marketing*. Deutscher Betriebswirte-Verlag, 1987.

[FKHF02] Stefan Fischer, Werner Kießling, Stefan Holland, and Michael Fleder: *The COSIMA Prototype for Multi-Objective Bargaining*. In proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002), Bologna, Italy, 2002, pp. 1364-1371.

[GL94] Terry Gaasterland and Jorge Lobo: *Qualified Answers That Reflect User Needs and Preferences*. In proceedings of the Twentieth International Conference on Very Large Databases (VLDB94), Santiago, Chile, 1994, pp. 309-320.

[GT00] Volker Gruhn and Andreas Thiel: *Komponentenmodelle – DCOM, JavaBeans, Enterprise JavaBeans, CORBA*. Addison-Wesley, 2000.

[Haf03] Bernd Hafenrichter: *Optimierung relationaler Präferenz-Datenbankanfragen*. Doctoral Thesis, University of Augsburg, 2003.

[Hai02] Matt Haig: *B2B E-Commerce Handbook: How to Transform Your Business-To-Business Global Marketing Strategy.* Kogan Page, January 2002.

[Han72] Flemming Hansen: *Consumer choice behavior – a cognitive theory*. The Free Press, 1972.

[Har02] Manfred Hardt: *Suchmaschinen entwickeln mit Java und Lucene*. JavaMagazin – Internet & Enterprise Technology, Volume 09/02, Software & Support, 2002.

[HEK03] Stefan Holland, Martin Ester, and Werner Kießling: *Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications*. In proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD '03), Dubrovnik, Croatia, September 2003, pp. 204-216.

[HK04] Stefan Holland and Werner Kießling: *Situated Preferences and Preference Repositories for Personalized Database Applications.* To be published in proceedings of the 23rd International Conference on Conceptual Modeling (ER2004), Shanghai, China, November 2004.

[HKP01] Vagelis Hristidis, Nick Koudas, and Yannis Papakonstantinou: *PREFER: a system for the efficient execution of multi-parametric ranked queries*. 2001 ACM SIGMOD international conference on Management of data, Santa Barbara, CA, USA, 2001, pp. 259-270.

[HL03] John Hunt and Chris Loftus: *Guide to J2EE: Enterprise Java*. Springer-Verlag, July 2003.

[Hol03] Stefan Holland: *Preference Mining and Preference Repository: Design, Algorithms and Personalized Applications*. Ibidem Verlag, Stuttgart 2004.

[Hor00] Albert S. Hornby: *Oxford Advanced Learner's Dictionary of Current English*, Cornelsen & Oxford University Press, October 2000.

[HP04] Vagelis Hristidis and Yannis Papakonstantinou: *Algorithms and applications for answering ranked queries using ranked views.* The VLDB Journal — The International Journal on Very Large Data Bases archive, Volume 13, Issue 1, January 2004, pp. 49-70.

[HS69] John A. Howard and Jagdish N. Sheth: *The Theory of Buyer Behaviour*. John Wiley and Sons Inc, New York, 1969.

[HS98] Stephen E. Heiman and Diane Sanchez: *New Strategic Selling: Unique Sales System Proven Successful By World's Largest Company's*. Warner Books, January 1998.

[Jil94] Nahid M. Jilovec: *The A to Z of Edi: The Comprehensive Guide to Electronic Data Interchange*. Independent Pub Group, October 1994.

[JSZC03] Rong Jin, Luo Si, ChengXiang Zhai, and Jamie Callan: *Collaborative Filtering with Decoupled Models for Preferences and Ratings.* In proceedings of the twelfth international conference on information and knowledge management, New Orleans, LA, USA, 2003, pp. 309-316.

[KBK03] Clare-Marie Karat, Jan Blom, and John Karat: *Designing personalized user experiences for eCommerce: theory, methods, and research.* In proceedings of the Conference on Human Factors in Computing Systems (CHI2003), Fort Lauderdale, FL, USA, April 2003, pp. 1040-1041.

[KF89] Robert Kass and Tim Finin: *The Role of User Models in Cooperative Interactive Systems*. International Journal of Intelligent Systems, Volume 4, 1989, pp. 81-112.

[KFD04] Werner Kießling, Stefan Fischer, and Sven Döring: *COSIMA^{B2B} – Sales Automation for E-Procurement*. In proceedings of the 6th IEEE Conference on E-Commerce Technology (CEC04), San Diego, CA, USA, July 2004, pp 59-68.

[KFHE01] Werner Kießling, Stefan Fischer, Stefan Holland, and Thorsten Ehm: *Design and Implementation of COSIMA - A Smart and Speaking E-Sales Assistant*. In proceedings of the Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2001), San Jose, CA, USA, 2001, pp. 21-30.

[KH02] Werner Kießling and Bernd Hafenrichter: *Optimizing Preference Queries for Personalized Web Services.* In proceedings of the IASTED International Conference on Communications, Internet and Information Technology (CIIT 2002). St. Thomas, Virgin Islands, USA, November 2002, pp. 461-466.

[KHFE01] Werner Kießling, Stefan Holland, Stefan Fischer, and Thorsten Ehm: *COSIMA - Your Smart, Speaking E-Salesperson*. 2001 ACM-SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 2001, p. 600.

[KHFH01] Werner Kießling, Bernd Hafenrichter, Stefan Fischer, and Stefan Holland: *Preference XPATH: A Query Language for E-Commerce*. 5. Internationale Tagung Wirtschaftsinformatik 2001, Augsburg, September 2001, pp. 427-440.

[Kie02] Werner Kießling: *Foundations of Preferences in Database Systems*. In proceedings of the Twenty-eighth International Conference on Very Large Data Bases, Hong Kong, China, 2002, pp. 311-322.

[Kie04] Werner Kießling: *Preference Constructors for Deeply Personalized Database Queries*. Technical Report 2004-7, University of Augsburg, 2004.

[KIW02] Hyeokki Kwon, Il Im, and Bartel Van de Walle*: "Are You Thinking What I Am Thinking?" – A Comparison of Decision Makers' Cognitive Maps by Means of A New Similarity Measure*. In proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS02) - Volume 3. Big Island, Hawaii, USA, January 2002.

[KK02] Werner Kießling and Gerhard Köstler: *Preference SQL - Design, Implementation, Experiences*. In proceedings of the Twenty-eighth International Conference on Very Large Data Bases, Hong Kong, China, 2002, pp. 990-1001.

[KKTG95] Gerhard Köstler, Werner Kießling, Helmut Thöne, and Ulrich Güntzer: *Fixpoint Iteration with Subsumption in Deductive Databases*. Journal of Intelligent Information Systems (4), 1995, pp. 123–148.

[KLS02] Oliver Kelkar, Joerg Leukel, and Volker Schmitz: *Price Modeling in Standards for Electronic Product Catalogs Based on XML*. In proceedings of the 11th International World Wide Web Conference 2002 (WWW 2002). Honolulu, Hawaii, USA, May 2002, pp. 366-375.

[Kni89] Bromley Kniveton: *The Psychology of Bargaining*. Athenaeum Press Limited, 1989.

[Kum02] Ram Kumar: *XML Standards for Global Customer Information Management.* DM Review, May 2002.

[LD99] Ting-Peng Liang and Her-Sen Doong: *Effect of bargaining in electronic commerce*. In proceedings of the International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 1999), Santa Clara, CA, USA, April 1999, pp. 174-181.

[LK02] Achim Leubner and Werner Kießling: *Personalized Keyword Search with Partial-Order Preferences*. In proceedings of the 17th Brazilian Symposium on Database Systems in Cooperation with ACM SIGMOD (SBBD 2002). Gramado, Brazil, October 2002, pp. 181-193.

[MA01] Yanping Ma and Esma Aïmeur: *Intelligent Agent in Electronic Commerce – XMLFinder*. In proceedings of 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2001), Cambridge, MA, USA, June 2001, pp. 273-278.

[MBDFH02] Scott McGlashan, Dan Burnett, Peter Danielsen, Jim Ferrans, Andrew Hunt, Bruce Lucas, Brad Porter, Ken Rehor, and Steph Tryphonas: *Voice Extensible Markup Language (VoiceXML) Version 2.0.* http://www.voicexml.org, 2002.

[MO99] Tsuyoshi Moriyama and Shinji Ozawa: *Emotion recognition and synthesis system on speech*. In proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS1999), Volume 1, Florence, Italy, June 1999, pp. 840-844.

[MZFLZ01] Jian Ma, Quan Zhang, Zhiping Fan, Jiazhi Liang, and Duanning Zhou: *An Approach to Multiple Attribute Decision Making Based on Preference Information on Alternatives.* In proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS01) - Volume 3. Maui, Hawaii, USA, January 2001.

[Nic66] Francesco M. Nicosia: *Consumer Decision Process*. Prentice-Hall, Inc., NJ, USA, 1966.

[ÖFA01] Hubert Österle, Elgar Fleisch, and Rainer Alt: *Business Networking – Shaping Collaborations between Enterprises*. Springer Verlag, Berlin, 2001.

[Pal00] Klaus Palme: *eCl@ss – das Klassifikationssystem für E-Commerce im Internet*. Leistung und Lohn, Zeitschrift für Arbeitswirtschaft – BDA. Heider-Verlag, June 2000.

[Rac89] Neil Rackham: *Major Account Sales Strategy*. McGraw-Hill Trade, January 1989.

[Rai90] Howard Raiffa: *The Art and Science of Negotiation*. Harvard University Press, July 1990.

[RAMV02] Francesco Ricci, Bora Arslan, Nader Mirzadeh, and Adriano Venturini: *ITR: a Case-Based Travel Advisory System*. In proceedings of the 6th European Conference on Case Based Reasoning (ECCBR 2002), Scotland, 2002, pp. 613-627.

[RD97] Ehud Reiter and Robert Dale: *Building Applied Natural Language Generation Systems*. Journal of Natural Language Engineering, volume 3, number 1 1997, pp. 57-87.

[Rei02] Matthias Reichelsdorfer: *Implementierung eines autonomen, personalisierten E-Shopping Agenten*. Diploma Thesis, University of Augsburg, 2002.

[RL02] Gerhard Raab and Nicole Lorbacher: *Customer Relationship Management*. Sauer-Verlag, 2002.

[RN96] Byron Reeves and Clifford Nass: *The Media Equation: How People Treat Computers, Televisions, and New Media as Real People and Places*. CSLI Publications, September 1996.

[Rou96] Daniel Rousseau: *Personality in computer characters*. Working notes of the AAAI-96 workshop AI/ALife, AAAI Press, Menlo Park, CA, USA, 1996.

[RSG01] Gustavo Rossi, Daniel Schwabe, and Robson Guimarães: *Designing personalized web applications*. In proceedings of the tenth international conference on World Wide Web (WWW 2001), Hong Kong, Hong Kong, May 2001, pp. 275-284.

[RZ97] Gabriel A. Radvansky and Rose T. Zacks: *The Retrieval of Situation-Specific Information*. Cognitive Models of Memory, 1997, pp. 173-213.

[SB94] Carl P. Simon and Lawrence Blume: *Mathematics for Economics*. W.W. Norton & Company, April 1994.

[Sch01] Marc Schröder: *Emotional Speech Synthesis: A Review*. In proceedings of Eurospeech 2001 – 7th European Conference on Speech Communication and Technology, Aalborg, Volume 1, 2001, pp. 561-564.

[Sed98] Tod A. Sedbrook: *A Collaborative Fuzzy Expert System for the Web*. The DATA BASE for Advances in Information Systems – Summer 1998, Volume 29, Number 3, 1998.

[Shi97] Neil Shister: *10 Minute Guide to Negotiating*. Hungry Minds Inc., 1997.

[SKPRH01] Volker Schmitz, Oliver Kelkar, Thorsten Pastoors, Thomas Renner, and Claus Hümpel: *Spezifikation BMEcat Version 1.2*. http://www.bmecat.org, 2001.

[Smi00] John R. Smith: *Interoperable Content-based Access of Multimedia in Digital Libraries*. In proceedings of the DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries, Zurich, Switzerland, 2000.

[Smi01] John R. Smith: *MPEG-7 Standard for Multimedia Databases*. In proceedings of the 2001 ACM-SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 2001, p. 627.

[Tho23] William I. Thomas: *The Unadjusted Girl*. Little, Brown, and Co. Boston, 1923.

[WBKH04] Qiuyue Wang, Wolf-Tilo Balke, Werner Kießling, and Alfons Huhn*: P-News: Deeply Personalized News Dissemination for MPEG-7 based Digital Libraries.* To be published in proceedings of the 8th European Conference of Digital Library (ECDL04), Bath, United Kingdom, September 2004.

[WCP02] Min Wang, Yuan-Chi Chang, and Sriram Padmanabhan: *Supporting Efficient Parametric Search of E-Commerce Data: a Loosely-Coupled Solution.* In proceedings of the 8th Conference on Extending Database Technology (EDBT 2002*),* Prague, Czech Republic, March 2002, pp. 409-426.

[WD02] Günter Wöhe and Ulrich Döring: *Einführung in die Allgemeine Betriebswirtschaftslehre*. Vahlen, 2002.

[Wie03] Julius Wiedemann: *Web Design - Taschen's 1000 Favorite Websites*. Taschen, Germany, 2003.

[XAF03] Bin Xiao, Esma Aïmeur, and José Manuel Fernandez*: PCFinder*: *An Intelligent Product Recommendation Agent for E-Commerce*. In proceedings of the 2003 IEEE International Conference on Electronic Commerce (CEC 2003), Newport Beach, CA, USA, June 2003, pp. 181-190.

[Yag03] Ronald R. Yager: *Fuzzy logic methods in recommender systems*. Fuzzy Sets and Systems, Volume 136, Issue 2, June 2003, pp. 133-149.

[Zad73] Lotfi A. Zadeh: *The Concept of a linguistic variable and its application to approximate reasoning*. Elsevier Pub. Co., New York, 1973.

[Zad83] Lotfi A. Zadeh: *A computational approach to fuzzy quantifiers in natural languages.* Computers and Mathematics with Applications, Volume 9, 1983, pp. 149-184.

List of Figures

Figure 1.1 Amazon’s failing search engine..9
Figure 1.2 Quality of search results at ACM 11
Figure 1.3 Not personalized e-procurement..12
Figure 1.4 B2B customer pyramid - source: ITSMA..12
Figure 1.5 Fields of research of this thesis..14
Figure 2.1 E-graph for Bart's EXPLICIT preference..20
Figure 2.2 Hierarchy of base preference constructors 22
Figure 2.3 ‘Better-than’ graph for Lisa’s Pareto preference..23
Figure 2.4 ‘Better-than’ graph for Homer's prioritized preference..23
Figure 2.5 ‘Better-than’ graph for Barnie’s numerical preference..24
Figure 2.6 Sample Preference XPath statement..26
Figure 2.7 Sample Preference Repository..27
Figure 2.8 Meta model for situations..28
Figure 3.1 Workflow of a BMO result presentation 32
Figure 3.2 Sample E-graph for an EXPLICIT preference 50
Figure 3.3 ‘Better-than’ graph for a sample $LAYERED_m$ preference 52
Figure 3.4 A nested complex preference example..73
Figure 3.5 Quality valuations of base preferences for result t_1 76
Figure 3.6 Number of ‘sufficient‘ valuations of a sample BMO^{++} 79
Figure 3.7 Sample frequencies of quality terms..82
Figure 3.8 Quantity of single characteristics with equal quality to the overall quality..84
Figure 4.1 Workflow of an automatic offer composition..92
Figure 4.2 Preference based composition of the shopping cart..95
Figure 4.3 Sample preference quality XML file for one BMO search result..96
Figure 4.4 Sample discounts as flexible XML based structure..99
Figure 4.5 Personalized and preference based price fixing..100
Figure 4.6 Structure of the BMEcat tag <ARTICLE>..102
Figure 4.7 Example of BMEcat Article_Reference interdependencies 103
Figure 4.8 eCl@ss characteristics for clear stock boxes..104
Figure 4.9 Price fixing by means of BMEcat data import 104
Figure 5.1 Architecture of the COSIMA application server..106
Figure 5.2 Sample customer’s search preferences..108
Figure 5.3 Smart search result presentation..108
Figure 5.4 End of a successful bargaining session..109
Figure 5.5 Product search and presentation settings..110
Figure 5.6 Global bargaining adjustments 111
Figure 5.7 Time effort of preference search and presentation 112
Figure 5.8 Time effort of the Preference Presenter..112
Figure 5.9 Selections regarding the presentation order of task 1..113
Figure 5.10 Selections regarding the presentation orders of task 4 113
Figure 5.11 Number of bargaining iterations..114
Figure 5.12 Savings gained via bargaining..115
Figure 6.1 Smily valuation of search results..118
Figure 6.2 Parametric search adjustments at Infineon 119
Figure 6.3 Sample result of Infineon's parametric search..119
Figure 6.4 Linguistic quantifier "at least α %" 121
Figure 6.5 Membership functions for fuzzy values of attribute weather 121
Figure 6.6 Query relaxation guided by ITR..124

List of Tables

Table 2.1 XML database excerpt of available cities 26
Table 3.1 Sample BMO result set 30
Table 3.2 Sample database excerpt 33
Table 3.3 Range of values for the linguistic variable WEATHER CONDITION 36
Table 3.4 Sample excerpt of a stock exchange database 39
Table 3.5 Sample qualities of notebook characteristics for one result tuple t..................................60
Table 3.6 Sample qualities of notebook characteristics for one result tuple t..................................62
Table 3.7 Sample qualities of notebook characteristics in ascending order.....................................63
Table 3.8 Qualities of notebook characteristics for sample tuple t_1 64
Table 3.9 Qualities of notebook characteristics for sample tuple t_2 64
Table 3.10 Sample qualities referring to a prioritized preference...65
Table 3.11 Sample excerpt of a stock exchange database 68
Table 3.12 Sample BMO result set for a complex nested preference 75
Table 3.13 The single quality values of Q of a sample BMO^+ for a complex nested preference 77
Table 3.14 Sample of overall qualities of a BMO result set 81
Table 3.15 Sample arbitrary search result set 88
Table 4.1 Sample product bundle of audio CDs 98

Appendix A

The DTD for the preference quality XML structure:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Title:         Preference Quality
    Description:   Document Type Definition for quality valuations
                   of BMO search results with respect to the search
                   preferences
    Version:       1.0
    Date:          18.12.03
    Author:        Stefan Fischer
    Copyright:     Stefan Fischer Copyright (c) 2003 Chair for
                   Databases and Information Systems, University of
                   Augsburg
-->

<!--
    PREFERENCEQUALITY is the quality valuation root for
    BMORESULTSETS.
-->

<!ELEMENT PREFERENCEQUALITY     (BMORESULTSET)*>

<!--
    BMORESULTSET is a container for BMORESULTS. For each result of
    the result set the single result qualities for each base
    preference and accumulated preference are declared.
-->

<!ELEMENT BMORESULTSET          (RESULT)*>

<!--
    Each result tuple is identified by the unique id. E.g. in
    BMEcat this is the Supplier_AID. Yet, this could also be
    specified much more complex with session id, situational
    context etc.
-->

<!ELEMENT RESULT                (QUAL)>
<!ATTLIST RESULT                id CDATA #REQUIRED>
```

```
<!--
    A preference quality QUAL describes the quality for a search
    result with respect to the search preference. In case of an
    accumulated preference the quality is recursively defined,
    based on the combined preference qualities, which are elements
    of this QUAL. QUALs of type base of course have no further
    elements, all other types at least two.

    The attributes of a QUAL are:
        ATTRIBUTE               POSSIBLE VALUES

        type:                   "base"
                                "complex"
        subtype:                for complex:
                                 "pareto"
                                 "prioritized"
                                 "numerical"
                                for base:
                                 "around"
                                 "between"
                                 "highest"
                                 "lowest"
                                 "score"
                                 "at_least"
                                 "at_most"
                                 "explicit"
                                 "pos"
                                 "pospos"
                                 "posneg"
                                 "neg"
                                 "layered"
                                 "contains"
        value:                  "perfect"
                                "very good"
                                "good"
                                "acceptable"
                                "sufficient"
        characteristic:         depends on the domain, e.g. the
                                price, make or something
        level:                  in QUALs with type
                                prioritized, there the
                                preference level will be declared,
                                e.g. 1 for the first level, 2 for
                                the second and so on, for the
                                involved preference
```

```
        distance:                for numerical base preferences -
                                 the distance to the optimal value
        dist_direction:          '+' or '-' for distance from or to
                                 the desired value(s) of a BETWEEN
                                 or AROUND preference
        measuring_unit:          e.g. cm or pieces
-->

<!ELEMENT QUAL                     (QUAL)*>
<!ATTLIST QUAL                     type CDATA #REQUIRED>
<!ATTLIST QUAL                     subtype CDATA #REQUIRED>
<!ATTLIST QUAL                     value CDATA #REQUIRED>
<!ATTLIST QUAL                     characteristic CDATA #REQUIRED>
<!ATTLIST QUAL                     level CDATA #IMPLIED>
<!ATTLIST QUAL                     distance CDATA #IMPLIED>
<!ATTLIST QUAL                     dist_direction CDATA #IMPLIED>
<!ATTLIST QUAL                     measuring_unit CDATA #IMPLIED>
```

Appendix B

The DTD for the flexible discount framework:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
    Title:            Discount Framework
    Description:      Document Type Definition for electronic data
                      interchange of complex product bundle
                      discounts
    Version:          1.0
    Date:             28.12.03
    Author:           Stefan Fischer
    Copyright:        Stefan Fischer Copyright (c) 2003 Chair for
                      Databases and Information Systems, University
                      of Augsburg
-->

<!--
    This discount framework provides a standardized structure for
    GENERAL DISCOUNTs of a product bundle.
-->

<!ELEMENT GENERALDISCOUNTS     (DISCOUNT)*>

<!--
    A DISCOUNT is the main construct for a single discount
    information. The attributes are:

  name:                  the name of the discount, e.g. regular
                         customer discount
  level:                 the calculation level, which is decisive
                         for the calculation basis, where this
                         discount is subtracted from
  value:                 the numeric value of the discount, e.g. 5
  valueunit:             the unit of the value, e.g. € or %
  valuetype:             one of the following kinds of calculation
                                "absolute"
                                "relative"
                                "user_defined_condition"
  condition:             the condition which is decisive, whether
                         this discount will be applied or not, e.g.
```

```
                        ">1000€" for a quantity discount for more
                        than 1000€ or to be registered as regular
                        customer for the regular customer discount
  conditionparameters:  e.g. the boolean value of being a regular
                        customer
-->

<!ELEMENT DISCOUNT          EMPTY>
<!ATTLIST DISCOUNT          name CDATA #REQUIRED>
<!ATTLIST DISCOUNT          level CDATA #REQUIRED>
<!ATTLIST DISCOUNT          value CDATA #REQUIRED>
<!ATTLIST DISCOUNT          valueunit CDATA #REQUIRED>
<!ATTLIST DISCOUNT          valuetype CDATA #REQUIRED>
<!ATTLIST DISCOUNT          condition CDATA #IMPLIED>
<!ATTLIST DISCOUNT          conditionparameters CDATA #IMPLIED>
```

Curriculum Vitae

Personal Data

Name:	Stefan Fischer
Date of birth:	15.11.1971
Place of birth:	Zusmarshausen, Germany
Nationality:	German
Marital status:	Unmarried

School and Occupational Career

09/78 – 08/82	Elementary school, Horgau
09/82 – 08/84	Secondary school, Zusmarshausen
09/84 – 08/88	Junior high, Neusäß with graduation
09/88 – 01/91	Professional Training with banker graduation at the Deutsche Bank AG Augsburg
02/91 – 06/92	Employee at the Deutsche Bank AG Augsburg
09/92 – 08/94	Upper vocational school, Augsburg with graduation
10/94 – 07/99	Studies of mathematics and economics at the University of Augsburg with graduation to the degree Dipl.-Math. oec.
since 10/99	Research assistant at the chair for databases and information systems at the University of Augsburg

Augsburg, 10.08.2004