Mobile Search in a Preference World

Werner Kießling Wolf-Tilo Balke
Institute of Computer Science
University of Augsburg
Augsburg, Germany
+49 +821 598 2136
{kiessling, balke}@informatik.uni-augsburg.de

ABSTRACT

We investigate the importance of focused search for mobile web-based services. Focused search aims at delivering only those query results to a mobile client that are best matches wrt. the current location, situation and individual preferences of the mobile user. Our approach is characterized by an intuitive and powerful notion of preferences, which are modeled as strict partial orders under a 'I like A better than B' semantics. The underlying theory of preferences, their implementation and integration into databases and Internet services has been investigated for several years within the research program *Preference World* at the University of Augsburg. For the scope of this paper we discuss some of its results with special attention to the topic of mobile search and we will outline ongoing research work. In particular we demonstrate how to do preference engineering for focused search applying the database query languages Preference SQL and Preference Xpath. The performance aspects of mobile services are addressed by architectural considerations and by a novel algorithm called SR-Combine for efficient top-k search. We show how techniques for progressive delivery, which change the delivery order preferring most relevant document parts within a given level of detail, can improve mobile search. Moreover, advanced personalization in mobile services will require to pay an even closer attention to the key notion of preferences, taking cognitive modeling aspects into account. In summary we claim that careful preference modeling and efficient preference-based queries are crucial for the success of mobile search and personalized mobile services.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information search and retrieval – *retrieval models, query formulation, search process.*

General Terms
Management, Performance, Design, Human Factors.

Keywords
Personalized Mobile Access, User Profiling, Cooperative Databases, Preference SQL, Preference Engineering, Preference XPATH, Focused Search.

1. INTRODUCTION

Appropriate focused search capabilities are an essential problem when designing mobile services. Due to low bandwidths and small display sizes, it is reasonable to return only a few result objects. However, these should have a high precision, because discarding irrelevant result objects using mobile devices is a very tedious task. To avoid the flooding of users with irrelevant objects, the paradigm of **top-k querying** has proven to be a valuable feature in today's applications. Users specify the number *k* of objects to return and get only the *k* best matching objects from the application. However, this requires a **cooperative answer behavior** [1] delivering the best matching objects only, even if no exact matching objects were found. Thus also the well-known 'empty result effect' can be avoided.

But of course the problem is how to determine the best matching objects with respect to a user's intuitive notion of relevance. For this problem we propose a new **preference-based retrieval model** that allows users to specify their queries in a qualitative and thus intuitive manner. Recently preferences are catching wide-spread attention in the software community, in particular in terms of personalization for B2C or B2B e-services. Here especially **user profiles, location-based information** and **domain knowledge** are integrated into the query process.

After some best-matching objects have been determined efficiently, they have to be delivered to the user. For this task appropriate **service architectures** for a variety of mobile clients have to be implemented using efficient components for querying, aggregating information and delivering the result documents. Especially for online information services, a **progressive delivery** of documents with media types specifically adapted for each client device's capabilities is desirable. We will show how to design a delivery component that automatically adapts digital content in generic XML formats for a suitable delivery to a variety of client devices. Finally we will focus on techniques for advanced personalization and investigate some factors having a strong influence on the search process.

2. PREFERENCE ENGINEERING FOR FOCUSED SEARCH

Generally speaking personalized searches show different facets: There is the 'exact world', where user wishes can be satisfied completely or not at all. Database queries in this context are characterized by hard constraints, delivering only objects exactly matching the constraints, otherwise rejecting the user's request.

But in real world applications personal preferences have to be understood in the sense of **wishes**. However, there is no guarantee that wishes can be satisfied at all times. In case of failure for a perfect match people are usually prepared to accept worse alternatives or compromises. This is true in particular for mobile situations, where a decision should be made rather quickly, e.g. think of a mobile restaurant search.

A careful examination of the nature of preferences in the real world reveals that they share a fundamental common principle [2]. It turns out that people express their wishes frequently in terms like "**I like A better than B**". This kind of preference modeling is universally applied and intuitively understood by everybody. Thinking of preferences in terms of 'better-than' has a very natural counterpart in mathematics: a preference can be formulated as **strict partial order** on a set of attribute names with an associated domain of values, which figuratively speaking is the 'realm of wishes'. Recently, [16] has been advocating a similar approach towards preference modeling.

But before we study a few examples, let us look at the unsatisfying state of the art: In today's e-services too often no or no reasonable answer is returned, though one has tried hard completing query forms to match one's personal preferences closely. Especially in mobile services including PDAs, WAP or i-mode phones the case of repeatedly receiving empty query results turns out to be extremely disappointing to the user, because it is not only tedious to reformulate a query several times, but also mobile services costs are often charged with respect to time.

Now consider a typical query in e-commerce applications:

Example1:

Get the best offers for compact hi-fi systems including a CD player in a price range of 100 to 115$.

Figure 1: Using a Preference SQL WAP service

Figure 1 shows a result delivered via a WAP gateway by a search engine based on Preference SQL [3]. Preference SQL implements a **Best Matches Only (BMO)** query model, which is obviously an appealing choice for e-services, especially for mobile search: The first query delivers already the best possible results only. In our case a perfect match to the user query was not available. But instead of delivering the empty result or an excess of irrelevant hits, Preference SQL returns only the best matching alternatives as shown above. Clearly such a behavior is preferable over other widespread ad hoc approaches to focused search like e.g. parametric search.

In general decisions are not based on a single preference, but on a complex combination of preferences. Let us consider another application example:

Example 2:

An insurance company is storing their letters, e-mails, contracts and related documents on a central server. At each damage event the person in charge wants to know about recent similar cases or the same persons involved in different cases. Thus a query has to be performed like: "Give me the top cases preferably recently with most similar kinds and costs of damages and same people involved."

In this typical example we can already identify four **basic preferences** (kind, costs, people, time) and rank documents by their relevance towards these issues. This ranking requires to build a complex preference from the basic preferences. To assemble **complex preferences** three construction operators (that can of course also be used in combination) have been proven to be especially useful in today's applications [2]:

- **Pareto accumulation** of preferences retrieves all those objects that are better or equal to other objects with respect to every single preference. The intuitive semantics of Pareto accumulation is a non-discriminating combination of *equally important* preferences.
- **Prioritization** evaluates preferences according to a user defined order. The intuitive semantics of prioritization is to evaluate *more important* preferences first and only refer to less important preferences, if the values for more important preferences are equal. An example for this kind of evaluation is the lexicographical order.
- **Numerical ranking** of preferences calculates *score values* for each database object and sorts objects according to their scores. In contrast to the qualitative Pareto accumulation and Prioritization this is a *quantitative approach* which is already quite common in today's database systems.

Reconsidering example 2 our insurance company could want to focus more on those people involved rather than on the kind of damage , time or costs. Then a prioritization of people over a Pareto accumulation of the other three issues would be the adequate complex preference expressing the company's needs in a very intuitive and easy to handle way.

With these definitions of base preferences and constructors for complex preferences we are ready to pose queries in a very intuitive manner, in particular if the non-numerical constructors are employed. But performing the proper preference modeling to compose the query to be issued is only one important constituent of a focused search. The second crucial part concerns the answer semantics of the query engine. A crucial design feature of our preference modeling technique is that preference construction is closed under the strict partial order semantics. Thus for every preference the notion of maximal elements is precisely defined. A 'Best Matches Only' BMO semantics retrieves exactly all maximal elements of a preference. In this way the notorious empty results, which are even worse in mobile applications, are avoided. But also the unwanted flooding effect can be controlled effectively by combining the virtues of BMO and of a general top-k retrieval model: If all BMO objects are less than k, some nonmaximal objects may be delivered as well. If it's greater than k, k objects may be selected (randomly) for display at the mobile device.

To include a BMO query semantics within worldwide SQL standard the commercial product **Preference SQL** has been implemented [3]. It has been tested extensively in e-commerce applications for comparison shopping [4, 5]. For the use in XML documents our sophisticated preference semantics has also been implemented in the XML query language **Preference XPath** [7]. In the following we will take a closer look at the features of Preference SQL.

Preference SQL compatibly extends standard SQL by introducing new language constructs that treat preferences in the form of partial orders as first class citizens. Preferences can be constructed on the fly when issuing a query, or they can be defined as persistent objects using a Preference Definition Language. Up to now standard SQL allows only for the specification of hard constraints in the WHERE clause. In Preference SQL **soft constraints** are syntactically expressed inside an SQL query block following the new keyword PREFERRING. In general, the Preference SQL query block offers the following options, allowing for hard and soft selections to co-exist within one single query:

SELECT	<selection>
FROM	<table_references>
WHERE	<where-conditions>
PREFERRING	<soft-conditions>
GROUPING	<attribute_list>
BUT ONLY	<but_only_condition>
ORDER BY	<attribute_list>

The elements that extend standard SQL appear in bold. Like in standard SQL, the WHERE and ORDER BY clauses are optional. Without a PREFERRING clause, it is not a preference query. The GROUPING (performing with soft constraints what GROUP BY does with hard constraints) and BUT ONLY (performing a quality control: tuples that are considered too bad to appear in the result set can be excluded directly) clauses are both optional. Preferences only apply to tuples fulfilling the WHERE condition. The condition of the BUT ONLY clause is logically tested after applying the preferences of the PREFERRING clause.

Reconsidering example 1, the Preference SQL query might look as follows, where 'AND' expresses Pareto accumulation:

```
SELECT description, price
FROM  devices
WHERE category = 'hi-fi systems'
PREFERRING description CONTAINS 'CD'
        AND price BETWEEN(100, 115)
```

Of course such a Preference SQL query is not supposed to be entered directly by a mobile user, instead it has to be generated automatically from a query form entered by the user.

The answer semantics of Preference SQL logically implements the Best Matches Only (BMO) query model:

- Find all perfect matches with respect to preference P in the `PREFERRING` clause. If this set is non-empty, we are done.
- Otherwise consider all alternative values, but discard worse values with respect to P on the fly. All non-dominated values are returned (modulo the BUT ONLY constraints).

Note that in case of Pareto accumulation BMO returns exactly the Pareto-optimal set. The SQL extension using a SKYLINE clause proposed in [3] is a subset of Preference SQL.

Preference SQL is implemented as an intermediate layer between the application and the database. It processes preference queries by rewriting them to standard SQL queries and submitting them to the database. Queries without preferences are just passed through to the database system without causing any noticeable overhead. Legacy SQL applications run without any restriction.

3. PERFORMANCE ISSUES IN MOBILE SERVICES

3.1 Service Architecture

With the current developments of devices like cell phones or PDAs, mobile services will play an important role in future information technology. Pervasive access to information becomes more and more attractive not only for business work, but also for private uses. The research area of data integration over the web has lead to several architectures for different applications [8, 9, 10]. Basically architectures for mobile services are mainly twofold:

- **Central Server Architecture (CSA):** If the service provider is also the content provider, services are provided using a high performance server with central data repositories.
- **Distributed Sources Architecture (DSA):** If the service provider and content provider are different or short update ranges are necessary, services may be provided using an application server gathering information on demand from distributed external data sources.

Enabling queries in mobile environments, however, poses some severe efficiency problems. In [10] ways to enable top-*k* querying for mobile services with a distributed sources architecture providing direct access to various Internet sources are presented. Though some of these services (e.g. location-based city maps or restaurant guides) would be desirable, tests show that the processing times even for simple tasks often need minutes to even hours. But psychology teaches that users only tend to accept response times up to 3 seconds before their questions are answered. This real-time restriction can generally be applied to online search engines. Users will allow higher run-times only for very difficult tasks (e.g. in desk top work environments), where they strongly depend on the results. Of course mobile users definitely do not want to wait an hour for on-demand real-time services like mobile traffic information or restaurant guides.

When trying to meet **real-time requirements** with today's bandwidths, a central server architecture is necessary. A solution to combine Internet sources with local database servers is given by [9] handling accesses to Internet sources in an asynchronous manner. Results are cached for later use in a central database server. Since the service provider obviously knows which type of queries to expect, what data is commonly accessed and how often updates

updates are needed, a caching strategy with asynchronous updates is well suited for most mobile services.

Figure 2 outlines the intended architecture. The mobile service resides on an application server providing a query engine to process preference queries, a combining engine to get the top *k* results sets efficiently and a delivery engine that prepares complex documents for (progressive) delivery. All digital content is retrieved from a local server that is updated in an asynchronous manner. As shown in a prototypical implementation [11] for the example of mobile online auctions our delivery engine is able to transform e.g. generic XML formats using XSLT to support any mobile device e.g. via WAP or i-mode gateways (cf. section 4). Another advantage of this architecture is that all data on the local server can be indexed to suit the design of the service. Through statistical analysis also the costs for certain usage patterns can be estimated providing the cost estimations to determine the ratio between different kinds of access personalized for each user.

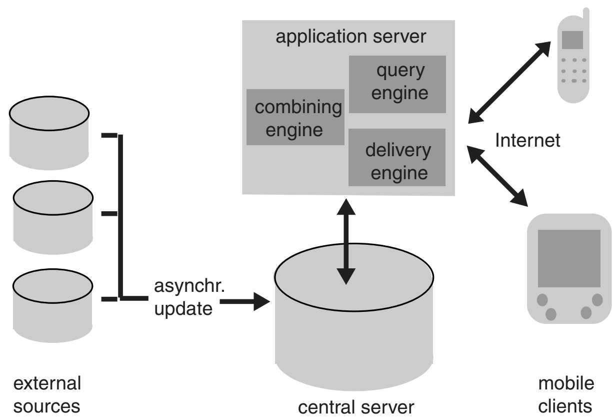

Figure 2: CSA architecture for mobile services

3.2 Performance Issues

To implement a top-*k* query model for middleware applications a variety of approaches has been presented recently [10, 12, 13]. All of them rely on numerically ranked preferences so far, attempting to minimize the number of database objects that have to be accessed before being able to deliver a correct result set. We just argued that for mobile applications the performance requirements even become harder. Though algorithms have to return only a few objects (the parameter k is typically rather small for mobile access), they have to be tuned well to a central server CSA architecture as just claimed, taking IO- and CPU-costs into account to meet stringent mobile real-time response time requirements.

Our approach towards this problem proposes the **SR-Combine** algorithm [14] overcoming the disadvantages of existing algorithms by suitably self-adapting to a variety of environments and controlling the run-time costs. SR-Combine optimizes the numbers and specific costs for object accesses and thus leads to short run-times. In particular it allows the earliest possible output of objects, while the algorithm is still running. Especially in mobile environments this is a useful feature, because the available bandwidth can be used more efficiently. Besides even in cases where the psychological 3 seconds barrier cannot yet be met, the successive output behavior is important. If an object is delivered every couple of seconds, psychologically users will not notice the waiting period until all requested objects have been returned. Thus though the total running time may exceed 3 seconds, the mobile application requirements can psychologically still be met.

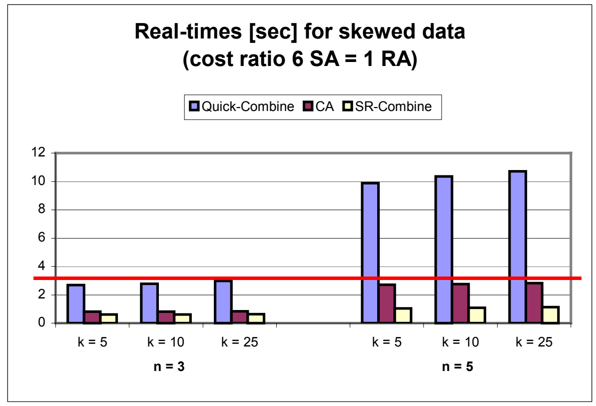

Figure 3: Benchmark of SR.-Combine's Runtimes [sec]

In a set of synthetic benchmarks simulating different mobile application scenarios, the efficiency of SR-Combine (in contrast to [12] and [13]) was shown to meet the psychologically founded barrier of 3 seconds response times in many practical cases for medium database sizes of 50000 objects for 3 and 5 preferences involving even skewed data (figure 3). But [14] shows also ways to deal with cases where the strict requirements are not yet met.

4. PROGRESSIVE CONTENT DELIVERY

With the convergence of Internet technology and mobile communication the bandwidth for e-services will significantly increase in near future. The current trend towards the mobile Internet even points to a wide acceptance of flexible, adaptable and personalized multimedia applications in public and private life. But digital content for mobile applications cannot be handled like common multimedia documents. Every document delivered has to take the client devices capabilities and technical limitations into account.

But also the user's interaction with digital content changes: for instance users in mobile environments will not be interested to wait for a complete document (possibly including media objects like images, etc.) to be delivered, but will expect techniques for **progressive delivery** changing the delivery order in a way that prefers most relevant document parts within a given level of detail. Thus users should be able to choose their relevance profile at a certain level of detail and get the most relevant document parts first. This would enable them to decide during data transmission whether their information need is already satisfied and the transfer can be aborted or some more document parts are needed.

For mobile searches in [15, 11] we propose a content selection investigating the progressive delivery of digital content in mobile environments. **Content selection** is concerned with selecting most relevant document parts for preferred delivery and choosing most

Figure 4: Progressive delivery of an XML newspaper article

appropriate data types and qualities of media objects to meet the user's profile. Major tasks of content selection are the **personalized adaptation** of digital content for specific applications and a progressive delivery of documents under technical constraints. Our approach towards content selection uses **XML** as a powerful tool for both structure description and semantic annotation. Due to a variety of established standards and tools XML has proven to be well suited for the implementation of value adding Internet services.

To make the content selection as effective as possible we exploit three main information sources:

- **Basic user preferences** stating their specific notion of relevance
- **Structural characteristics** of each document together with semantic author annotations
- **Usage profiles** and **technical constraints** of mobile client devices (e.g. memory or display size)

For delivery the complex XML document has to be broken down to single blocks with a certain level of detail. These blocks form the smallest units of information and must be delivered without further division. By definition XML documents form tree-like structures. Basically these documents consist of mark-up nodes, which are assembled inductively to document trees. Our approach towards relevance includes textual and structural properties to get an overall satisfying result.

However, using the structural semantics via author annotations provides an additional benefit: annotations can be used by service providers to support specific usage patterns. Anticipating these patters can essentially improve the quality of service. Consider for instance the restricted Internet capabilities of a modern cell phone. Due to its rather small display, low resolution and low bandwidths it may be preferable to delay or even suppress the delivery of images. Using appropriate **style sheets** and **XSLT transformations** a selection of generic XML documents can be adapted automatically to the user's needs and the client device's technical profile.

An example of progressive delivery is given in figure 4. Using the CNN Sports Golf News an article has been automatically transformed from simple HTML via XML into the final WML delivery format and delivered to a Palm Pilot V. As can be seen the subheading of the article has been chosen to be most relevant as it contains the significant information (in contrast to the heading 'Sub-70 Sensation'). Structural elements like agency, date, and location are chosen to be significant, too. The article starts with the most important paragraph, whereas the image is delivered last.

After the XML documents have automatically been transformed, a newly reordered document is synthesized most suitable for delivery. However, practical tests of the above concept in the area of mobile Web have shown that when reordering document parts the content's readability often is severely affected and if low levels of detail for the reordering are chosen, documents may even become completely incomprehensible. Thus in [15] we proposed measures to detect fragmentation and also developed strategies for defragmentation to improve the documents' readability. **Defragmentation** in this case tries to improve the reordering (that is based on relevance measures only) by moving previously adjacent text parts that have been separated closer together. Obviously there is a trade-off between restoring the order and preferring the most relevant document parts. In our experiments the strategies managed to improve the documents readability significantly while maintaining the early delivery of most relevant document parts.

5. TOWARDS ADVANCED PERSONALIZATION IN MOBILE SERVICES

To pursue a narrowly focused search in mobile environments some more considerations on adequate preference modeling are necessary. We showed how users can intuitively state their preferences, how preference queries can be processed efficiently and how the result content can be delivered with respect to structural and technical constraints. But how queries can automatically be engineered that are most suitable for the application domain and current mobile situation has not yet been addressed.

As a basic principle cognitive sciences [4] state that although user interaction with retrieval systems will generally be purposeful, the user will in general not be conscious of all the necessary means to be taken. Thus in order to raise its performance in terms of relevance, a retrieval system has not only to focus on explicit user specification, but should also take information into account which is known about the user's situation or behavior. This information can mainly be gathered from four sources:

- **Long-term preferences**: The notion of relevance from previous retrievals is used (cf. section 2).
- **Intention**: The specific user's purpose of the retrieval is included in searches.
- **Situation**: The present state and environment of a user is included in the retrieval process.
- **Domain**: Knowledge on the specific domain is used.

For effective retrieval knowledge from all these sources has to be blended with the specific user-provided details and keywords. Thus the query should be expanded with **cognitive heuristics** and information. For this query expansion again no typical database selections can be used, because due to the exact match retrieval model, heuristic information could simply overwrite user-provided information and/or the results set could rapidly get empty by too many hard constraints. So the intended expansion has to be implemented using **soft constraints** that may be relaxed step by step until there are definitely objects matching the expanded query. In essence this is again the cooperative answering capability offered by Best Matches Only languages like Preference SQL or Preference Xpath.

Besides techniques for user modeling for the application in **focused mobile search** especially the situation and domain knowledge is important. Consider e.g. a user's color preferences in e-commerce applications. Basically those preferences may be important when purchasing a car, etc., but in domains like bookstores color preferences can hardly make sense. Similarly, special user-specific profiles may demand a procedure deviating from standardized queries and retrieval processes. For instance, personalized newspapers and information services have been developed recently that gather information about the relevance or level of interest of each article from the user's place of residence. Of course, these heuristics will definitely fail if somebody is specially interested in worldwide news or the development of certain topics given by the long-term preferences.

For specific situations, e.g. specified by usage patterns, the query model thus has to be adapted. Similar to necessary **context awareness** in agent technology, information systems often have to deal with constraints imposed by the current situation. Consider for example time: depending on the situation there generally is a trade-off between retrieval quality and response times [6]. If only a short time is given to make a decision, one would want to trade the size or accuracy of result objects for faster response times. Another example for focused search strongly influenced by the situation are **location-based services** that use GPS data to get the user's location. This information is for instance helpful for city guides, where the exact part of the plan that is needed can be delivered or for restaurant guides where recommendations for preferred restaurants near the current location can be retrieved. As a last example even the **emotional state** of the user may contribute to a high precision search. Modern agent negotiation technology for instance already makes use of psychological user profiles or the specific user's mood. In ubiquitous computing projects even mobile services for health care using medical data have been researched. How all of this can be integrated into a **cognitive query builder** for automatic preference-based query expansion is a challenging area of forthcoming research. In this context especially XML and ontologies like in [17, 18] will be most promising technologies for design and implementation.

6. CONCLUSION

We have given insights into recent research and development results from the research initiative 'Preference World' at the University of Augsburg, seen from the perspective of mobile search. One main objective of 'Preference World' is to improve search engine technology in order to build truly personalizable information systems. One central issue towards the solution of this goal is the proper way to deal with the notion of preferences. We argued that preference modeled as strict partial orders are an appealing choice: they have an intuitive semantics, they may be subjective from daily life experiences, driven by personal intentions, or due to technical constraints. Our extensible preference model both unifies and extends existing approaches for non-numerical and numerical ranking and opens the door for a systematic approach towards preference engineering. This preference model has been implemented already on top of standard query languages, namely Preference SQL for the world of SQL databases and Preference Xpath for the XML world. Preference queries exhibit the important 'BMO'-property, meaning that only best matches with respect to the user preferences are retrieved.

Preference-based search technology under a BMO query semantics can successfully cope with the notorious empty-result and flooding problems of many search engines, enabling a comfortable focused search that gets the desired information (and nothing else) already by the first user query. We argued that in particular for mobile services such a focused mobile search is mission-critical. Precision of query results being one crucial point, efficiency of mobile search is another one. We have addressed performance aspects of mobile services by architectural considerations, arguing that central server architectures are essential for psychologically acceptable response times. Our novel algorithm SR-Combine is optimized for such architectures and can efficiently process top-k searches for numerically ranked preferences. Given a wide range of practical query profiles, it can obey a 3-second response-time barrier which is particularly important for mobile applications. Moreover, we showed how techniques for progressive delivery, which change the delivery order preferring most relevant document parts within a given level of detail, can improve mobile search. Our prototype implementation of progressive delivery can be set up automatically using current XML technology, hence supporting a wide range of portable devices including PDAs, WAP cell phones and I-mode cell phones.

In summary we claim that careful preference modeling and efficient preference-based queries are crucial for the success of mobile search and personalized mobile services. Relying on commercial products and industry standards Preference World has put forward already a set of powerful technologies to build personalized information systems. But definitely, still more efforts must be undertaken. For instance, we have pointed out that advanced personalization in mobile services will require to pay an even closer attention to the key notion of preferences, taking cognitive modeling and socionic aspects into account.

Current and forthcoming preference research at the University of Augsburg focuses on advanced personalization technologies for (mobile) services and includes the following projects:

- ‘P-NEWS’, funded by the German Research Society DFG, investigates preference engineering and the design of preference repositories to a MPEG7-based digital library application. (www.informatik.uni-augsburg.de/lehrstuehle/info2/projekte/p-news/)
- ‘COSIMA’, funded as part of the Bavarian Research Association FORSIP on Situation-based, Individualized and Personalized Man-Machine Interaction, investigates preference-based e-negotiation ([5], www.mycosima.com).
- A heuristic query optimizer for Preference XPATH, exploiting the preference algebra from [2], is being developed, funded by DFG as part of the Research Group ‘Efficient Electronic Coordination in the Service Sector’.
- Preference mining is supposed to become part of the ‘ProMine’ project with several participating European partners, focusing on user modeling.
- The cognitive query builder will be investigated as part of a forthcoming cooperation with University of California, Berkeley within the ‘Data Centers’ project.

7. REFERENCES

[1] J. Minker: An overview of cooperative answering in databases. Intern. Conference on Flexible Query Answering Systems, LNCS 1495, Springer, 1998

[2] W. Kießling: Foundations of preferences in database systems, Tech. Report 2001-8, Institute of Computer Science, University of Augsburg, Germany, 2001 (http://www.informatik.uni-augsburg.de/skripts/techreports/)

[3] S. Börzsönyi, D. Kossmann, K. Stocker: The skyline operator, Intern. Conference on Data Engineering (ICDE), Heidelberg, Germany, 2001

[4] R. Ackhoff, F. Emery: On Purposeful Systems, Aldine Atherton, Chicago, 1972

[5] W. Kießling, S. Fischer, S. Holland, T. Ehm: Design and implementation of COSIMA - a smart and speaking e-sales assistant, WECWIS, San Jose, CA, USA, 2001, pp. 21-30

[6] D. Grossman, O. Frieder: *Information Retrieval: Algorithms and Heuristics*, Kluwer Publishers, 1998.

[7] W. Kießling, B. Hafenrichter, S. Fischer, S. Holland: Preference XPATH: a query language for e-commerce, 5. Internationale Tagung Wirtschaftsinformatik, Augsburg, Germany, 2001, pp. 43-62

[8] S. Gribble, E. Brewer: System design issues for internet middleware service: deductions from a large client trace. Proc. of the USENIX Symposium on Internet Technologies and Systems, Monterey, CA, USA, 1997

[9] R. Goldman, J. Widom: WSQ/DSQ: A practical approach for combined querying of databases and the web. ACM Conf. on Management of Data (SIGMOD), Dallas, TX, USA, 2000

[10] N. Bruno, L. Gravano, A. Marian: Evaluating top-k queries over web-accessible databases. Intern. Conf. on Data Engineering (ICDE), San Jose, CA, USA, 2002

[11] M. Wagner, W-T. Balke, W. Kießling: XML-based multimedia middleware for mobile online auctions, Intern. Conference on Enterprise Information Systems (ICEIS), Setubal, Portugal, 2001, pp. 934-944

[12] U. Güntzer, W-T. Balke, W. Kießling: Optimizing multi-feature queries for image databases, In Proc. of the International Conference on Very Large Databases (VLDB 2000), Cairo, Egypt, 2000, pp. 419-428

[13] R. Fagin, A. Lotem, M. Naor. Optimal aggregation algorithms for middleware. ACM Symposium on Principles of Database Systems (PODS), Santa Barbara, CA, USA, 2001

[14] W.-T. Balke, U. Güntzer, W. Kießling: On real-time top k querying for mobile services, Tech. Report 2002-2, Institute of Computer Science, Univ. of Augsburg, Germany, 2002 (http://www.informatik.uni-augsburg.de/skripts/techreports/)

[15] M.Wagner, W. Kießling, W.-T. Balke: Progressive content delivery for mobile e-services, Tech. Report 2002-3, Institute of Computer Science, Univ. of Augsburg, Germany, 2002 (http://www.informatik.uni-augsburg.de/skripts/techreports/)

[16] J. Chomicki: Querying with intrinsic preferences, Proc. of the Intern. Conference on Advances in Database Technology (EDBT), Prague, Czech Republic, 2002, pp. 34-51

[17] I. Horrocks: DAML+OIL: a reason-able web ontology language, Proc. of the Intern. Conference on Advances in Database Technology (EDBT), Prague, Czech Republic, 2002, pp. 2-13

[18] A. Theobald, G. Weikum: The XXL search engine: ranked retrieval of XML data using indexes and ontologies, Proc. of the Intern. Conference on Modeling of Data (SIGMOD), Madison, Wisconsin, USA, 2002