Ontology-supported Preference Handling for Mobile Music Selection

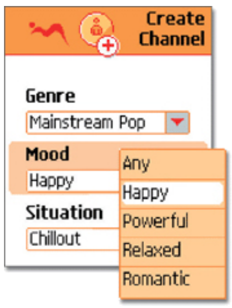
Olaf Noppens[1] and **Marko Luther**[2] and
Thorsten Liebig[1] and **Matthias Wagner**[2] and **Massimo Paolucci**[2]

Abstract. Mobile music selection is becoming an emerging market. So far only little work has been spent in selection of semantically enriched music streams on tiny mobile devices. This paper discusses how ontology-based preferences can be used for the selection of mobile music and presents a service discovery toolbox which is responsible for service matchmaking and preference relaxation. On the basis of this toolbox we developed a ubiquitous user interface using a graphical service discovery paradigm.

1 Introduction

With the ongoing improvement of wireless networks, bandwidths, and mobile client device capabilities, the number of available services offered to mobile users are being boosted by providing many different types of content from music to news to location-based and navigation services. Hence, mobile operators need to meet the challenge that on the one hand many users are overwhelmed with an increasing amount of service offerings. In addition, as in the last decade the number of mobile users has significantly arisen, the operators try to satisfy the individual users by providing as much as possible different service types to best match the individual wishes. On the other hand, mobile users want to maintain control of the service offerings, and individually select services according to their individual needs. One currently highly emerging example of mobile services is the vision of accessing universal radio and music access through the mobile Web from anywhere, anytime. A mechanism to select between different Internet Radio and music providers is crucial because already today, popular Internet radio stations host thousands of live streams [8, 6]. In this context, personalized access to audio content is particularly important to accommodate both, varying technical as well as personal user needs and preferences: Web radio stations offers audio content as virtual audio files which are streamed over the Web in different formats and compression rates for a wide range of different types of mobile media devices. Moreover, the heterogeneity of users with different taste of music makes it difficult to offer the same radio stations for all users – or at least a subgroup of users. Even a user likes different kind of music at similar situations (e.g. at traveling to work every morning) because there are not measurable parameters such as its mood etc. To handle these contrasts, users require a way to express their preferences on the type of services that they would like to receive and on the aspects of these services. In this paper, we present the implementation of an ontology-based preference paradigm that allows users to express individual preferences about the services they are interested in, and to support a personalized mobile service discovery which makes use of relaxing user preferences in the case that the most specific one(s) cannot be fulfilled. We introduce our semantic-based service discovery toolbox in the content of mobile music selection. The toolbox consists of two components, namely MOBIONT and MOBIXPL. The first one represents a service discovery server and supports personalized discovery with the help of plug-able relaxation strategies. The MOBIXPL mobile application allows users to explore relevant service categories and to express their preferences in a graphical way. In this paper we mainly focus on details about the service discovery based on the relaxation of user preferences. A detailed description of the technical part of the toolbox and of an earlier prototype of MOBIXPL introducing its novel user interface for preference building can be found in [11].

Currently, technology trends for distributing audio content on the Web are being brought from a pure desktop environment to the mobile web. A first product using some personalization techniques for the discovery of music stream is the StreamMan from Sony [9]. However, the personalization within Sony's StreamMan is limited to an explicit user ranking. Here, the user states whether he likes or dislikes already discovered music streams. In addition, as it can be seen in Figure 1 users can create in a graphical way so-called channels based on different predefined genres and moods. In cooperation with the mobile service provider Vodafone, Sony has announced to commercialize the StreamMan (which is now called “Vodafone Radio DJ”).



(a) Creating a channel

(b) Simple preferences

Figure 1. Sony's StreamMan for personalized music streaming

[1] Ulm University, Faculty of Computer Science, Ulm, Germany, email: olaf.noppens@uni-ulm.de, thorsten.liebig@uni-ulm.de
[2] DoCoMo Communications Laboratories Europe GmbH, Munich, Germany, email: luther@docomolab-euro.com, wagner@docomolab-euro.com, paolucci@docomolab-euro.com

The remainder of this paper is organized as follows. Section 2 introduces the mobile internet radio service scenario as our running example. Section 3 deals with ontology-based preferences and with preference modeling. In Section 4 we discuss how cooperative service discovery utilizes relaxation and we introduce several relaxation strategies. Section 5 gives a short overview of the implementation of the presented preference paradigm in our toolbox and on the mobile end-user interface.

2 Mobile Internet Radio

In this paper, we study ontology-based user preferences and personal service discovery on mobile devices considering a fictitious mobile Internet radio scenario. Web radio stations typically distribute their audio content as streams in a legal fashion over the Web. They offer audio data in different formats (such as MP3 or Real) and different compression rates in order to make them accessible through a wide range of different media devices. In contrast to the traditional radio stations, Internet radio stations typically provide more specialized audio content and therefore they are often further differentiated through a variety of content characteristics, For instance, characteristics describing the program format (i. e. News, Music, Educational programs etc.) or the local region where the station's program is hosted can be found. As they are freely available, anybody with a streaming client can listen to these stations anywhere in the world.

In our case study, we assume that Internet radio stations can be modeled as Web services with varying service characteristics. These characteristics describe and classify radio stations in terms of their program format, location, audio format, and a time-based classification of streamed audio content.

The Mobile Internet radio scenario serving here as our running example is only one of many possible application for our service discovery toolbox. Just by adding location information the number of potential application domains is growing, e.g. leisure time activity planer, shopping guide, etc.

3 Ontology-based Preferences

Providing the adequate service to each individual user in its individual and specific situation will become more and more importing with the growing amount of audio streaming content and of potential users. We advocate that making an informed choice of the right service will essentially include matching the individual user preferences and dislikes against the services offered in a given situation or context and have introduced the concept of user-centered Web service discovery [1] and selection [2]. Based on this work, basic yet very intuitive user preferences can be defined, accumulated and taken into account during service provisioning.

In the implemented preference framework, user preferences are expressed with respect to an underlying service ontology. The purpose of such an ontology is to semantically describe and classify service instances with the help of several categories. In each category, similar aspects are grouped together. Service discovery is then performed via a query which combines aspects of the same and/or different categories via logical conjunctions. Figure 2 depicts an extract of our radio scenario ontology. Here, only the categories Program and Region, modeled as sub-ontologies, are partially shown. For instance, the Program category characterizes the program type of radio streams such as music channel streams, news channel streams or streams broadcasting information about sports. Aspects are modeled in terms of ontological concepts: Music branches into several specialized genres such as classical, jazz or pop music.

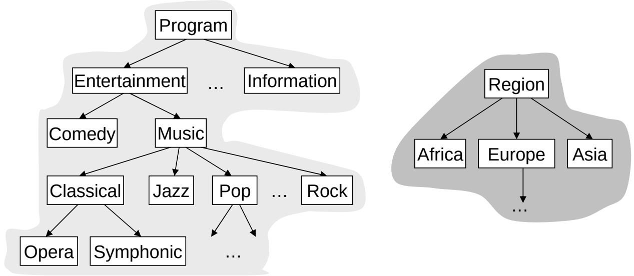


Figure 2. Extract from our sample radio scenario ontology showing categories about program type and regional information.

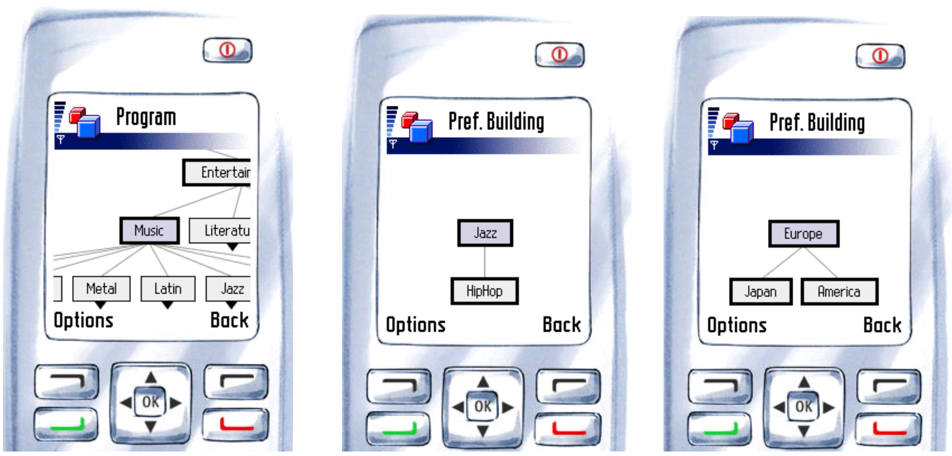
User preferences are then expressed in terms of service aspects orderings. In order to build preferences users can browse the ontology of each category in an intuitive way by gradually expanding or deexpanding the aspects. We think that such an on-demand expansion paradigm is very suitable for mobile devices with a limited screen display. Navigation is performed via the arrow keys of the phone keypad. Figure 3 (a) shows a sample screenshot of browsing through the category Program.

The idea of (partially) ordered aspect sets which are directly handled without the use of any explicit ranking or quality values was first introduced in preference algebra in the context of database systems proposed in the work of Kiessling [5]: here, preferences are expressed in terms of a special relation with the semantics of considering some aspect A superior to another aspect B. Informal spoken, this states that the user "likes A better than B". In addition, with the help of negative preferences, one can model statements to express that a user "does not like A".

To manage multiple user preferences complex preferences can be inductively constructed from a set of base preferences by means of preference constructors. In the following we will exemplarily discuss two basic combination operators, namely Pareto accumulation and preference prioritization. A formal definition of these two operators can be found in [5] where these operators among others have been first introduced. These two operators are currently implemented in the service discovery component of our toolbox. For the following, let assume that a user has defined preferences for the program as well as region category as shown in Figure 3 (b) and (c).

Pareto Accumulation In our case the Pareto-optimality principle is used to handle equally important user preferences. We use the binary Pareto operator $\otimes$ that is intuitively defined such that a matching feature tuple $v = (v_1, v_2)$ can only be preferred to another match $w = (w_1, w_2)$, if v is better than or equally good as w in every single feature value with 'strictly better' in at least one value.

Let us revisit the running example and consider the case where we want to further differentiate between the base preferences Program and Region. The aggregated preference $Program \otimes Region$ is depicted in Figure 4. As we discussed above, this preference consists of aspect combinations ordered according to the Pareto semantics. Therefore, the user's first choice consists of a radio station that broadcasts European jazz music followed by the next-best matches in stations with with lower relevance, e.g. American jazz or European hiphop music [4].



(a) Browsing the service ontology (b) Program category (c) Region category

Figure 3. Browsing the Internet radio ontology (a), and two sample Internet radio preferences.

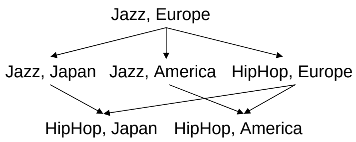


Figure 4. Pareto accumulation $Format \otimes Region$.

Preference prioritization Often some preferences might be considered more important than others. Intuitively, if $P_1 \& P_2$ (i.e. preference P_1 is prioritized over P_2) then P_1 is considered more important than P_2 [5]. As a consequence there is no compromise in feature matching by P_1. P_2 is evaluated only where P_1 gives several alternatives of equal usefulness, e.g., in the case of the choice between Japanese and American radio stations.

In our running example let us assume that – possibly due to technical constraints – a user wants to express that the encoding of a program is actually more important to him than its origin. Respecting the individual feature preferences for Program and Region, this prioritization can be stated in the single preference expression $Program \& Region$ as depicted in Figure 5.

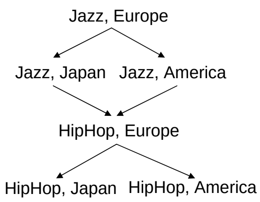


Figure 5. Preference prioritization $Program \& Region$.

Atomic Negation In many cases it might be important to express not only positive preferences but also to state the dislike of some aspects. Intuitively, this can work as a black-list filtering which eliminates all service instances which belong to the set of negated aspects. Note that such a set is considered as a hard constraint in a sense that it cannot be relaxed. To avoid inconsistent preferences the usage of atomic negation is limited. An aspect is not allowed in an atomic negation expression if it is used within a (non-negated, accumulated) preference expression. Otherwise a set of preferences could state that a user likes and dislikes for example jazz music. Due to the hierarchical nature of ontologies, this also means disallowing the negation of an aspect which is a super-aspect within an non-negated, accumulated preference expression.

As user preferences cannot always be fulfilled, prioritization and Pareto-based preferences have to be understood as whishes in terms of constraints that should be fulfilled. However, even if none of the indicated preferences are met, a service match can still be possible which not best meet the preference requirements as discussed in the next section.

4 Relaxation of User Preferences

In the proposed framework for personalized service discovery based on preferences, the preferences guide the service discovery to retrieve only those service instances which represent the best matches to a set of given user preferences. Hence, the aim of preferences is twofold. On the one hand, preferences are used to prune the set of all available services with respect to the set of preferences. On the other hand, as it is not always possible to fulfill the most specific preferences we now introduce the paradigm of relaxing the preference. In this section we will first introduce the basic mechanism of relaxation and then present and discuss some relaxation strategies which are currently implemented in our toolbox.

4.1 Basic Relaxation Strategy

The preference-based service discovery is performed along the lines of the given preference order. The service discovery starts at the most specific preference expression which best fits the user's wishes. If during service discovery no match could be found, the next step consists of gradually relaxing the expression to next-best matches. For that purpose, the aggregated preference relation is consulted to determine a relaxation path according to the user preferences. This idea is based on preliminary work found in the context of evaluating preference queries in database systems [5].

Formally, let $G_P = (R, E)$ be a relaxation graph which is build with help of the aggregated preference relation P. Let $R = R_0 \cup R_1 \cup \ldots \cup R_n$ be the set of the accumulated preferences [3] (see Section 3), and $E = E_0 \cup E_1 \cup \ldots \cup E_m$ the set of direct relaxation edges. Now we define R and E as shown in the following: R_0 is the set of base preferences, $R_0 = \{r | r\ is\ most\ specific\ preference\ expression\}$, $R_i = \{r_i | \exists r_j \in R_{i-1} : r_j\ P\ r_i\}$, $E_i = \{(r_j, r_k) | r_j \in R_{i-1}, r_k \in R_i\}$. Because P is a transitive relation, G_P might be a graph with transitive edges. From now on, we consider G_P as being non-transitive, i.e. the transitive relationships are eliminated from P. A query path Q_{G_P} is then a sequence $(v_0, v_1 \ldots v_i)$ of distinct sets of vertices of G_P such as $v_i \subseteq R_i$. Note that v_i is a set of accumulated preference expressions which means that a query path may be based upon several relaxation paths where v_0 contains all logical combined preference expressions. That is true in the case of indifferent preference expression as the preference relation P imposes an unordered set of expressions. W.l.o.g. we assume that G_P is the graph resulted after the application of atomic negation.

For instance, in our running example, the query path of the relaxation graph presented graphically in Figure 4 is (v_0, v_1, v_2) with the following sets: as there is only one base preference, v_0 solely consists of $\{\langle Jazz, Europe\rangle\}$. $v_1 = \{\langle Jazz, Japan\rangle, \langle Jazz, America\rangle, \langle HipHop, Europe\rangle\}$ and $v_2 = \{\langle HipHop, Japan\rangle, \langle HipHop, America\rangle\}$. If during service discovery no service instance could be found for the expressions in v_0, the next discovery step consists of trying to discover radio stations that broadcast Japanese or America Programs, i. e. stations that match at least one of the expressions within the set v_1.

In the case that no user-specific preferences are given or no further relaxation can be performed based on the user preferences (i. e. there is no further element in the query path), the relaxation can follow the domain-specific structure of the service ontology itself [1]. Such an ontology-based relaxation strategy relaxes the aspects which form the most specific preference expressions to the next general aspect(s) according to the subsumption hierarchy of the ontology. Ontology-based relaxation is based on the assumption that where a best match to the initial user preferences is not available, the user might also be satisfied with services belonging to more general aspects. Such a near match might still lay within the broader context of the original focus. Ontology-based relaxation can be treated as implicitly stated preference relations with the inverse subsumption relation as a preference, e. g. "A subsumes B" is modeled as "B is preferred over A".

4.2 Preference-related prioritization

As already mentioned in Section 3 preference prioritization is a strict linear ordering of preferences to state that some preferences are considered more important than other ones [5]. Even if it is desirable that a user explicitly prioritizes preferences in some situations, it seems reasonable to automatically activate preference prioritization and to derive a prioritization order esp. in the context of mobile devices in order to disburden the user from the additional task of explicit prioritization. Note however that, from a usability perspective, the user needs to be informed about automated prioritization.

[3] Note that the accumulated preferences of our prefer-over relations only allow at most one aspect of each category in R_i. W.l.o.g. we assume a lexicographic ordering of categories C_i such that the aspect within R_j can be ordered as following: $R_j = \langle a_0, a_1, \ldots a_n\rangle$ with $a_i \in C_i$.

The idea is that if a user defines preferences in a more fine-graded way within a category in comparison to another category, one can derive that the preferences of the first category are considered more important than those of the second one. Only if a strict ordering of all categories for which preferences are defined can be established, an automatic prioritization is possible. In a first step, we implemented a simple heuristic which compares the number of basic prefer-over preferences of each category. This heuristic is based on the observation/assumption that if a higher number of preferences related to one category are defined than related to another category this might be an indication that the user has a more detailed idea about the aspects of the first category and therefore consider it more important than the other category.

4.3 Delayed ontology-based relaxation

As already discussed, the ontological structure can also be used for relaxation. Specific service requests expressed in terms of aspects will be relaxed to more general requests based on the subsumption hierarchy of the ontology. Due to the hierarchical nature of ontologies services which are instances of some specific aspects are also (implicitly) instances of all more general aspects. In a naive ontology relaxation this might lead to an *under-specification* of the relaxation expression. As a consequence this may result in information flooding because services which are instances of a specific aspect as well as of one of its (direct or indirect) super-aspects are found. Formally, let $v_i = \left\{\left\langle v^0_{i,0}, \ldots, v^0_{i,n}\right\rangle, \ldots, \left\langle v^k_{i,0}, \ldots, v^k_{i,n}\right\rangle\right\}$ be one element of a query path Q_P. Then v_i is under-specified iff one accumulated preference expression is subsumed w.r.t. all of its aspects by another preference expression, i.e. iff $v^n_{i,j} \sqsubseteq v^m_{i,j}$ ($\forall j \in \{0, \ldots, n\}$, $n, m \geq 0, n \neq m$). The delayed ontology-based relaxation removed then all expressions which causes the under-specification, i. e. $v_i = v_i \setminus \cup v^m_{i,j}$. Thus, $v^m_{i,j}$ will be delayed until $v^n_{i,j}$ is relaxed via ontology-based relaxation. Note that it depends on the ontological structure in which element of the query path the delayed expression will occur.

Consider the case that a user is totally indifferent between radio stations broadcasting European opera music and jazz music and that no radio stations are currently performing these music types. Using ontology-based relaxation without delaying relaxation steps $\langle Opera, Europe\rangle$ is relaxed to $\langle Classical, Europe\rangle$, and $\langle Jazz, Europe\rangle$ is relaxed to $\langle Music, Europe\rangle$. Classical is a subaspect of Music as one can see in the program-ontology shown in 6. Therefore, the service discovery will consider all Music streams, not only classical resp. jazz music streams. In contrast, the delayed ontology relaxation delays the step which relaxes $\langle Jazz, Europe\rangle$ to $\langle Music, Europe\rangle$. Note that if no service instance is available for $\langle Classical, Europe\rangle$, this expression will be relaxed to $\langle Music, Europe\rangle$ which is the delayed expression from the previous relaxation step.

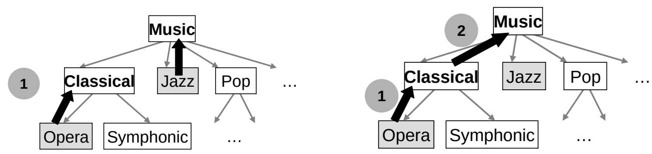
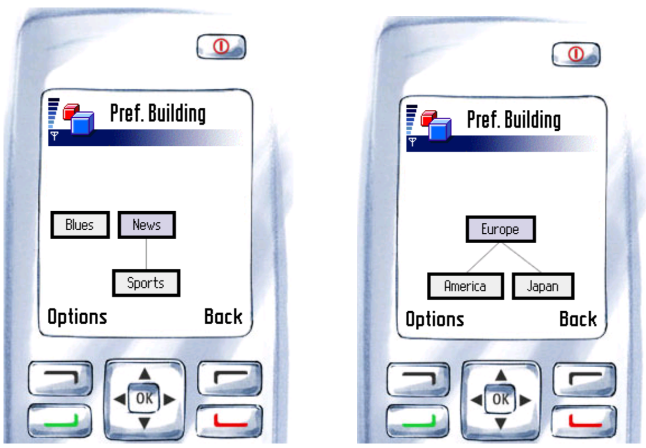


Figure 6. Ontology relaxation (a) without and (b) with delaying of relaxation steps.

4.4 Sequence-based querying

As mentioned in Section 3, the implemented preference framework uses the pareto accumulation principle to handle equally important user preferences. The more equally important user preferences exist, of the more expressions the elements of a query path G_P comprise. As a consequence, not only the search space for service discovery increases but also the potential matched services because there are several expressions in each relaxation step which may match service instances. Note that from the perspective of a strict understanding of the pareto accumulation principle, the result set of discovered services is complete and correct. To avoid the problem of information flooding in the context of our service discovery framework for mobile devices, we impose an ordering of the elements of each set v_i of the relaxation path G_P. For instance, defining preferences for the categories program and region as shown in Figure 7 leads to a relaxation Graph $G_P = (v_0, v_1, \ldots, v_n)$ with $v_0 = \{\langle News, Europe\rangle, \langle Blues, Europe\rangle\}$ and v_1 consists of $\langle Sports, Europe\rangle$, $\langle News, Japan\rangle$, $\langle News, America\rangle$, $\langle Blues, Japan\rangle$ and $\langle Blues, America\rangle$. Similar to the heuristic presented in the previous section for preference-related prioritization the analysis of the preferences can be used to state that the preferences of the category region are more important than those of the category program, one can simulate prioritization with a multi-set ordering. However, with additional background information about preferences a user has defined previously and about services a user has invoked, a more fine-graded ordering can be used (e. g. user preferences state that he is totally indifferent between Japanese and American radio programs but in the past he has always listen to Japanese radio programs). Note that such a strategy may prune the service instances at the risk of loosing completeness.



(a) Program category (b) Region category

Figure 7. Sample preferences.

5 Architecture of the Service Discovery ToolBox

Our service discovery toolbox consists of a server provider and discovery server, namely MOBIONT, and one or more requesters running the MOBIXPL client. Figure 8 depicts the architecture of the toolbox. Conceptually, MOBIONT serves as a semantic service portal implementing semantic service ontology lookup and service discovery. Here, user preferences are translated in matchmaking queries and these are then evaluated. Because there are not at any time services available which fulfill the most specific preference expression(s), the service discovery makes use of different kinds of preference relaxation strategies as discussed in the previous section. MOBIONT should be understood as a *service discovery testbed* in order to evaluate preference-based service discovery and to support the straightforward definition and deployment of different service discovery strategies for easy comparison and evaluation. For instance, our framework allows the definition and implementation of new strategies via its extensible plugin architecture at run-time. Moreover, the system supports the combination of several strategies to a linear chain of strategies to specify which one (if any) should be used if a previous one fails to return any service matches during discovery. Currently, we have implemented several strategies for service discovery using preference relaxation under Pareto- and Prioritization semantics (please refer to the previous section). Obviously, service discovery in the context of mobile environments is a cooperative process between the provider (or a third party portal) and the user. For instance, a requester needs to know which service aspects of a particular domain are available before he can specify any individual preferences. For the task of browsing the offered service properties and easily defining service requests in terms of personal preferences we implemented MOBIXPL as a graphical front-end to the mobile user as one can see on the right hand side of Figure 8. This client application provides a novel graphical interface for easy browsing of service ontologies and building user preferences. The sub-ontologies of each category are graphical represented in a graph-based structure according to the hierarchy. The user can then browse through the ontology by gradually expand or de-expand the aspects. In our opinion, such an on-demand expansion paradigm is very suited for mobile devices with a limited screen display because only user-relevant aspects are visible. Additional HCI techniques such as thumbnails for not-expanded aspects as well as usage pattern based on profiles (e.g. automatically expanding parts of an ontology, default preferences for different situations etc.) guide the user through the process of defining preferences.

Preference expressions are specified in two phases: selection of aspects and preference building. The interface is optimized to allow switching between both phases at any time, e.g. selecting some aspects, building a preference relationship between them, and then selecting other aspects. Selections and de-selections of the focus aspect within the graphical representation of the ontology can be made by pressing the "hash". key on the keypad. For each category, the user can separtly switch into the preference building mode. By default, all aspects are unbiased, i.e. considered as being equally important. To establish a new preference relationship the user navigates with help of the direction keys through all previous selected aspects while changing the focus aspect. If the focus lays on the aspect which should be the origin of a preference, the user presses the "hash" key and, in a second step, he changes the focus to the aspect which should be specified as the target of the preference relationship. Before MOBIXPL will establish a new preference relationship each user interaction is evaluated in order to avoid redundant definitions or inconsistencies.

For analyzing the discovery process in detail MOBIONT also provides a graphical tracking workspace. Here, one can manually or automatically step through every single processing state in order to gain in-deep information about the discovery outcome. This ranges from the identification why a preference could not be fulfilled to reconstructing the preference relaxation process in a graph-based visual-

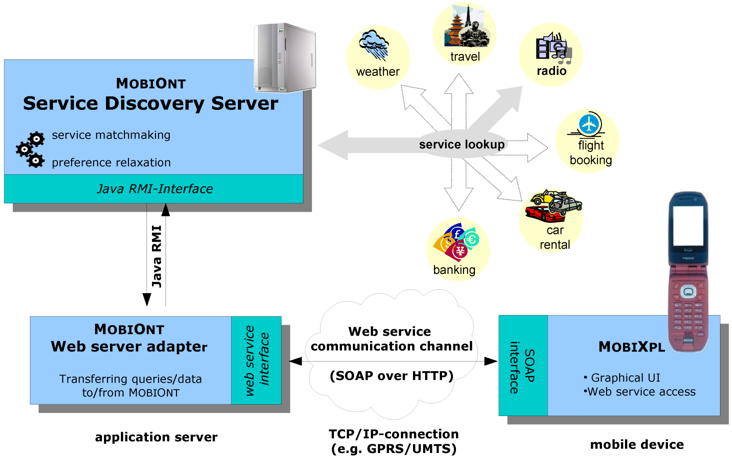


Figure 8. System architecture

ization component. It is possible to step back and forth through a history of successive discovery operations for a quick comparison of different relaxation strategies.

From an implementation perspective, MOBIONT provides a web service interface for interacting and communicating with client applications for a more flexible use. Note that is not limited to the mobile device client MOBIXPL but can be used from every application supporting Web service access. Therefore, MOBIONT consists of two components, namely the web server adapter running inside an application server and the service discovery server which performs the service discovery based on user preferences. These two components exchange data via Java RMI which allows the physical separation of both components (e.g. running the discovery server on a compute server). Recently we have ported MOBIXPL to a real mobile phone. Its novel graphical interface for easy browsing service ontologies and building preferences is an early adopter with respect to visualization technologies on mobile phones. MOBIXPL utilizes SVG-Tiny [10] which is a subset of the W3C standardized Scalable Vector Graphics targeted to resource-limited devices. Within J2ME it is an optional package which is specified as part of a Java Specification Request. SVG is a language for describing two-dimensional graphics and graphical applications in XML with the advantage of being compact in file size and scalable without quality loss. Both features are of benefit in a discovery scenario which aims at seamless browsing of possibly large category taxonomies on limited devices with varying display resolutions. Due to unavailability of a real device supporting SVG we are forced to use the official Nokia device emulator [7] for the Nokia N91 phone (S 60 3rd Ed. platform) which originally has been announced to be on the market by the end of 2005.

6 Summary and Outlook

The vision of a mobile and ubiquitous Web in which a growing number of different users access various kinds of information, anytime and anywhere, might soon become a reality. Finding the adequate service for a specific user in a certain situation is a demanding problem. We advocate that making an informed choice of the right services will essentially include matching the individual user preferences and dislikes against services offered in a specific situation or context.

In this paper, we have presented our work on ontology-based preferences and different relaxation strategies to fulfill user preferences. These ideas are built on preference algebra [5] and on our previous work on preference-based Web service discovery [1] and selection [2] enhanced through ontology views and search strategies [3] to effectively support the user in finding an appropriate service.

We have implemented the preference-based service discovery within our toolbox which is designed to serve as an open framework with well defined interfaces for straightforward integration and composition of alternative discovery mechanism within the context of user personalization in mobile environments.

Future work will be spent on extending the system towards enhanced personalization techniques and advanced user modeling such as location-based and situation-aware usage pattern. In addition, generic preferences could be learnt from observing user interaction in MOBIXPL. Another open issue is the ranking of discovered services according to observed user interaction and user behavior.

REFERENCES

[1] W.-T. Balke and M.Wagner, 'Cooperative Discovery for User-centered Web Service Provisioning', in *Proc. of the 1st Int. Conf. on Web Services (ICWS)*, Las Vegas, USA, (2003).

[2] W.-T. Balke and M.Wagner, 'Towards Personalized Selection of Web Services', in *Proc. of 12th Int. World Wide Web Conf. (WWW)*, Budapest, Hungary, (2003).

[3] W.-T. Balke and M. Wagner, 'Through different eyes – assessing multiple conceptual views for querying web services,', in *Proc. of 13th the Int. World Wide Web Conf. (WWW)*, New York, USA, (May 2004).

[4] B.A. Davey and H.A. Priestley, *Introduction to Lattices and Order*, Cambridge Mathematical Textbooks, 1990.

[5] W. Kiessling, 'Foundations of Preferences in Database Systems', in *Proc. of the 29th Int. Conf. on Very Large Data Bases (VLDB)*, pp. 311–322, Hong Kong, China, (2002).

[6] Radio Locator. http://www.radio-locator.com/.

[7] Nokia. Prototype SDK 4.0 for Java ME, December 2005.

[8] NullSoft SHOUTcast. http://www.shoutcast.com.

[9] Sony Streamman. http://www.streamman.net/.

[10] W3C. Mobile SVG Profiles: SVG Tiny and SVG Basic. Recommendation, January 2003. `http://www.w3.org/TR/SVGMobile/`.

[11] Matthias Wagner, Thorsten Liebig, Olaf Noppens, Steffen Balzer, and Wolfgang Kellerer, 'Towards Semantic-based Service Discovery on Tiny Mobile Devices', in *Proceedings of the Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications*, pp. 90–101, Hiroshima, Japan, (Novemer 2004).