Towards Semantic-based Service Discovery on Tiny Mobile Devices

Matthias Wagner[1], Thorsten Liebig[2], Olaf Noppens[2],
Steffen Balzer[2], and Wolfgang Kellerer[1]

[1] Future Networking Lab
DoCoMo Communications Laboratories Europe GmbH
Munich, Germany
{wagner,kellerer}@docomolab-euro.com

[2] Department of Artifical Intelligence
University of Ulm
Ulm, Germany
{liebig,noppens,balzer}@informatik.uni-ulm.de

Abstract. The vision of a mobile Web in which the computing environment will be composed of various devices that are carried by different users as they go through their daily routine might soon become a reality. In this context, Web services and the Semantic Web are already recognized as important building blocks. On the other hand, so far only little work has been done to support the discovery and selection of semantically enriched Web services on mobile terminals. In this paper we present a graphical toolbox for preference-based service discovery on tiny mobile devices. Our prototypical toolbox is developed as an extension of the Protégé OWL plugin and consists of a component called MOBIONT responsible for service matchmaking and preference relaxation together with an emulator for different mobile devices called MOBIXPL. Our work aims to serve as an open platform in order to gain experiences with different discovery strategies as well as mobile end-user interfaces.

1 Introduction

Among the most important resources of the future Web will be those that provide actual services. By ”service” we mean Web resources that do not merely provide static information but allow the user to effect some action or change in the world. In particular, the vision of the Semantic Web strongly depends on semantic-based services that enable users and software agents to discover, invoke, compose, and monitor network services. Using Web-based services has already become an integral part of our everyday life. Semantic Web technology and the advent of universal and mobile access to Internet services will only add to the broad range of existing services on the Web and provide additional features like knowledge-based, location- or context-aware information. In order to keep this increasing amount of (mobile) services manageable, it is essential that users are supported in the discovery and selection of services they personally need in a given situation or context.

We present work in progress towards MOBIONT – a semantic toolbox to explore mobile user-centered services on the Semantic Web. MOBIONT was only recently established as a collaborative project between NTT DoCoMo Euro-Labs and the Semantic Web group of the Department of AI at the University of Ulm. The vision of our joint work is to take full advantage of future complex service offerings on limited client devices and to handle the need for personalized service discovery in mobile environments. In this paper we study the case of a future mobile Internet radio scenario as a MOBIONT use case. Internet radio has become increasingly popular in recent years with boosting numbers in Web radio stations and subscribers. Platforms like SHOUTcast [10] allow people to easily set up new stations and broadcast audio content from their PC to listeners across the Internet or any other IP-based network. Already today, popular Internet radio portals host thousands of live streams [10,9]. The vision of universal radio access through the World Wide Web is further accelerated through the mobile Web with wireless access to audio content from anywhere, anytime. In this context, personalized access to content is particularly important to accommodate both, varying technical as well as personal user needs and preferences.

The rest of this paper is organized as follows. Section 2 introduces the radio service scenario as our running example and gives a first overview of MOBIONT. Our prototype builds on two major preliminary research efforts: previous work on ontology visualization and exploration is now being revisited in the context of our prototype and briefly described in section 3 . Section 4 surveys our efforts in preference modeling and discusses how cooperative service discovery is implemented to minimize user interaction. Section 5 features a detailed discussion of our prototypical implementation before section 6 concludes with summary and outlook on previous work.

2 A MOBIONT Case Study: Mobile Internet Radio Discovery

In this paper, we study the case of a future mobile Internet radio scenario. Internet radio is one of the currently highly emerging technology trends for distributing audio content on the Web in a legal fashion. Web radio stations are typically implemented as virtual audio files which are streamed or ”broadcasted” over the Internet. The audio data is sent in possibly different formats (such as MP3, Real or WMV) and compression rates to make Web radio stations accessible to people in a wide range of connections and through different types of media players. Very often, radio stations are further differentiated through their program format (News, Music, Educational program, etc.) and their origin, i.e. the local region where the station's program is created. Anybody with a PC that has audio capabilities and is connected to the Web can listen to these radio stations anywhere in the world.

As a MOBIONT proof of concept we study user guidance and service discovery in a fictitious wireless Internet radio portal. To this end we assume that Internet radio stations can be modeled as Web services with varying service characteristics. In our case study, we have build a basic Internet radio ontology (cf. ontology frame in Figure 1 and Figure 2) that consists of concepts to describe and classify Web radio services in terms of program format, origin, audio format characteristics and a time-based classification of streamed audio content. Figure 1 depicts the principal buildings blocks of MOBIONT

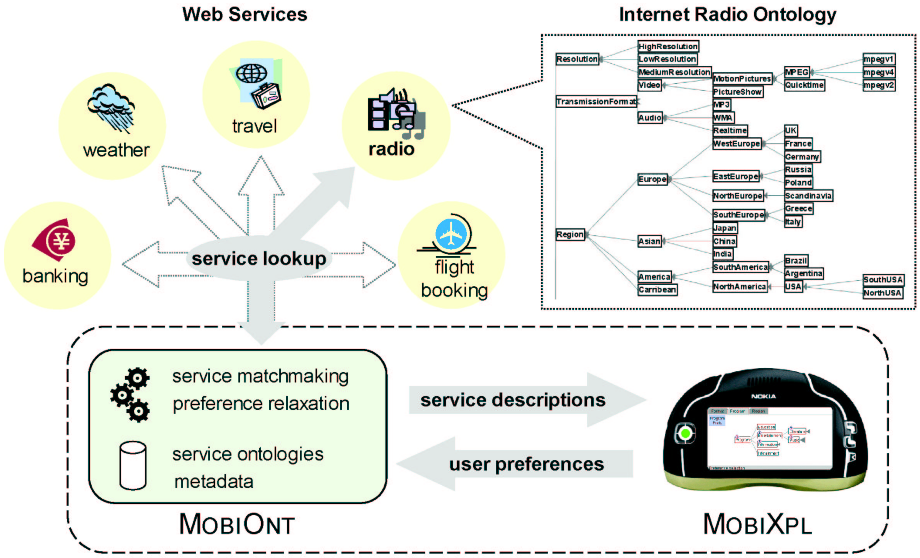

Fig. 1. Usage of MOBIONT in a wireless Internet radio scenario.

and illustrates how preference-based service discovery is implemented. Our Internet radio scenario is only one of many possible applications for the MOBIONT framework. Parts of the Internet radio ontology are carefully exposed to the user through MOBIXPL, the graphical frontend to our framework. MOBIXPL emulates different mobile terminals and consists of a mobile ontology browser with support for individual user views as well as an intuitive interface to user preferences. The idea is to only display selected concepts and sub-ontologies depending on the user's experience level and usage profile. While browsing the service ontology, concepts that circumscribe services with key relevance to the user can be selected and combined to user preferences. Subsequently, these preferences are used during the service discovery to implement cooperative behavior: if the search for a perfectly matching radio station fails, the initial query is gradually relaxed along the lines of the determined preferences until a next-best match can be found. Both application aspects, selective ontology browsing as well as preference building and mapping are explored in the remainder of this paper.

3 Ontology Browsing

To build Semantic Web applications for non-experienced end-users, adequate tools for building, visualizing, and exploring ontologies are needed. This demand for simple, flexible as well as intuitive user interfaces. On the other hand, most of todays ontology browsing interfaces only provide list-based representations for display and editing of ontologies. From a usability perspective, those interfaces inherently suffer substantial drawbacks concerning search and navigation efficiency as well as user orientation and effective ontology management. More advanced visualization techniques like tree maps or Venn diagrams still have shortcomings when it comes to ontology representation: they exploit nested graphical structures overlapped with edges that can obscure large portions of ontologies in many practical cases. Other rendering techniques, like hyperbolic tree visualization, have the disadvantage of ”fisheye distortion” which makes it difficult to read off-center labels and are therefore not suitable for low resolution displays.

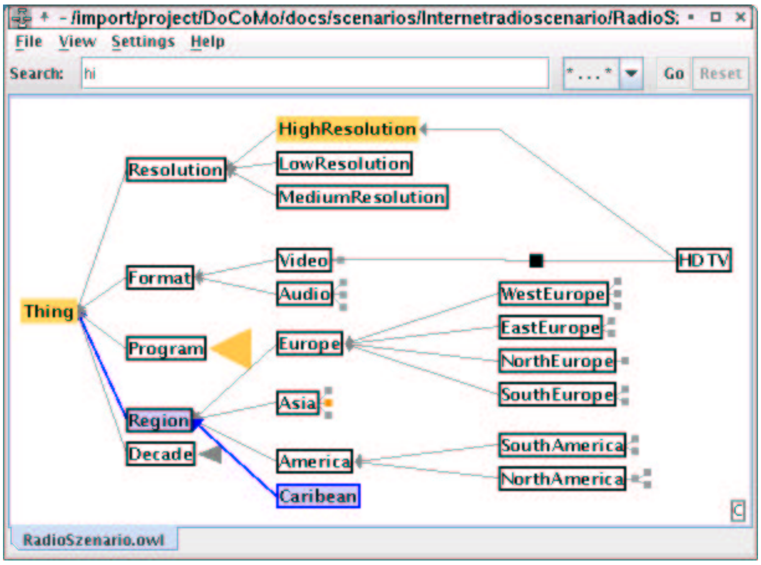

Fig. 2. ONTOTRACK ontology visualization and authoring tool.

We have implemented a novel approach to visual ontology management that combines hierarchical layout techniques with context sensitive zooming features. Furthermore, mouse enabled editing is optimized for navigation and manipulation of large ontologies. Our ontology authoring tool ONTOTRACK [7,8] provides animated expansion and de-expansion of class descendants, zooming, paning, and uses elaborated layout techniques like click-able miniature branches or selective detail views. Figure 2 shows a screen shot of ONTOTRACK that displays the partially expanded Internet radio ontology from our running example (shaded nodes represent search matches, shaded and boxed nodes indicated an inheritance path). ONTOTRACK always shows the semantically correct inheritance hierarchy to the user while trying to optimize the graphical representation. Consequently, ancestors of nodes are rendered as thumbnails if they have not been explicitly expanded (e. g. see ancestors of node HDTV in figure 2). Other ONTOTRACK features include instant search highlighting and drag-n-drop editing.

As a standalone application, ONTOTRACK was so far designed to be used on desktop-sized displays. In bringing parts of the ONTOTRACK functionality to MOBIXPL – the graphical frontend of MobiOnt – we have therefore tried to adapt the key visualization features for usage on small devices. As a result, MobiXpl layouts ontologies as directed acyclic graphs and allows their convenient navigation whereupon the charateristics of limited mobile terminals are respected: ontology layouts are continuously zoomed, centered or panned, nodes are expanded and de-expanded depending on the navigational path and display size. In addition MobiXpl provides additional graphical highlighting in order to supply the user with auxiliary information, e. g. icons indicate the existence of individuals, triangles and miniature trees approximate not expanded sub-branches, tool-tips show the number of node descendants.

4 Preference-based Service Discovery

Even though the capabilities of today's Web-based services are still relatively simple, their sophistication and diversification will grow with the improvement of (wireless) networks, bandwidths, and client device capabilities. Consequently, finding the adequate services will become a more and more demanding problem for the individual user. We advocate that making an informed choice of the right service will essentially include matching the individual user's 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 our preference framework, user preferences are expressed in terms of service feature orderings. As proposed for example in preference algebra [6], these (partially) ordered feature sets are directly handled without the use of any explicit quality or ranking values: user preferences are introduced as a special relation with the semantics of considering some object (or class) A superior to another object (or class) B ("I like A better than B"). Intuitively, during service discovery such user preferences have to be understood as wishes that should, however, cannot always be fulfilled. In that sense, preferences indicate constraints that a service should fulfill to best meet its requirements. On the other hand, even if none of the indicated preferences are met, a service match can still be possible. Figure 3 shows a set of sample preferences in our Music scenario. Here a user indicates that she generally prefers radio programs from Europe over those from Japan or America. Still, the latter two choices are her preferred choices over any other available program.

Preference/service matching is performed along the lines of the given preference order. Thus, if during service discovery no match could be found in European programs, the next discovery step consists of trying to match radio stations that broadcast Japanese or American programs. If neither of these two second-best choices is available, any other program is matched. The other base preferences in Figure 3 state that for the transmission format MP3 is generally preferred over Real. Furthermore the program preference expresses that for the program category News or Blues (a sub-category of Music) are considered first choices. The next best choice to News is Sports followed by Music. While News and Blues are treated as equally good choices, there is no direct means of comparing Sports and Blues.

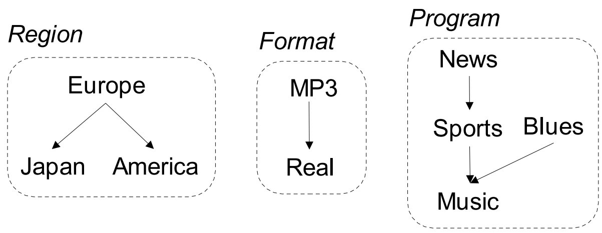

Fig. 3. Basic Internet radio preferences.

To manage multiple user preferences complex preferences can be inductively constructed from a set of base preferences by means of preference constructors. We will briefly discuss two basic combination operators, namely Pareto accumulation and preference prioritization, which are currently implemented in the service discovery component of MOBIONT. In the following, P is a preference defined as a strict partial order $P = (A, <P)$, where A is a set of service features and $<P \subseteq dom(A) \times dom(A)$ an order relation with $dom(A)$ as the domain of feature A. $<P$ is irreflexive and transitive.

4.1 Pareto Accumulation

The Pareto-optimality principle has been applied and studied intensively for decades for multi-attribute decision problems in the social and economic sciences. In our case it 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. Formally, let $P_1 = (A_1, <P_1)$ and $P_2 = (A_2, <P_2)$ be two preferences. For $x = (x_1, x_2)$ and $y = (y_1, y_2)$ with $x,\ y \in dom(A_1) \times dom(A_2)$ we define $x <(P_1 \otimes P_2)\, y \Leftrightarrow (x_1 <P_1\, y_1 \wedge (x_2 <P_2\, y_2 \vee x_2 = y_2)) \vee (x_2 <P_2\, y_2 \wedge (x_1 <P_1\, y_1 \vee x_1 = y_1))$. Therefore $P = (A_1 \cup A_2, P_1 \otimes P_2)$ is a strict variant of the coordinate-wise order of the Cartesian product [4] and is called a Pareto preference or the Pareto accumulation of P_1 and P_2.

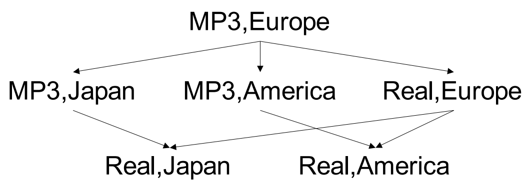

Fig. 4. Sample pareto accumulation $Format \otimes Region$.

Let us revisit the running example and consider the case where we want to further differentiate between the base preferences Region and Format. Using Pareto accumulation the aggregated preference $Format \otimes Region$ is depicted in Figure 4. As we discussed above, this preference consist of feature combinations ordered according to the Pareto semantics that tread the two base preferences equally. Therefore, the user's first choice consists of a radio station that broadcasts a European program encoded in MP3 followed by next-best matches in stations with lower relevance, e.g. American programs in MP3 or European programs in Real format.

4.2 Preference prioritization

Often some preferences might be considered more important than others. In addition to the equal treatment of preferences by Pareto accumulation, we define the preference operator $\&$ for preference prioritization. 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. 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. Formally, let $P_1 = (A_1, <P_1)$ and $P_2 = (A_2, <P_2)$ be two preferences. For $x = (x_1, x_2)$ and $y = (y_1, y_2)$ with $x,\ y \in dom(A_1) \times dom(A_2)$ we define $x <(P_1 \& P_2)\, y \Leftrightarrow (x_1 <P_1\, y_1) \vee (x_1 = y_1 \wedge x_2 <P_2\, y_2)$ and claim that this definition still holds if $x_1 \sim y_1$, i.e. in the case where x_1 and y_1 are incomparable. Therefore $P = (A_1 \cup A_2, P_1 \& P_2)$ is a strict variant of the lexicographic order of the Cartesian product [4] and is called a prioritized preference.

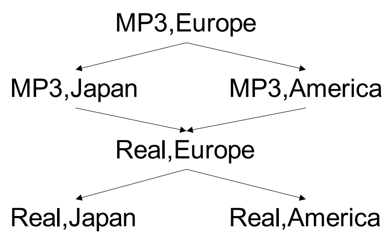

Fig. 5. Sample preference prioritization $Format\&Region$.

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 Format and Region, this prioritization can be stated in the single preference expression $Format\&Region$ as depicted in Figure 5.

5 Service Discovery Toolbox: Features and Implementation

We have implemented the system components that were briefly introduced in Section 2 as a first prototype of our MOBIONT toolbox. The prototype is developed as an extension of the Protégé OWL plugin and is fully integrated with Protégé. As a consequence, the full features of Protégé OWL plugin can be used in parallel to create and manipulate

MOBIONT service ontologies. Protégé is an open environment for knowledge acquisition and knowledge-based system written in Java and maintained at Stanford University [5]. Its open architecture supports the easy creation of customized plugins that bring added functionality to the standard Protégé environment.

The implementation of our demonstrator is driven by the objective of maximal flexibility with respect to graphical interface techniques as well as selection methods in order to gain experiences with different selection strategies and query relaxation techniques. Conceptually, our framework consists of the actual MOBIONT component that implements service ontology lookup and cooperative service discovery. The graphical frontend MOBIXPL to the portal provides ontology visualizations to the mobile user and allows the definition of service requests in terms of personal preferences (cf. Figure 1). Technically, the main components of the prototype are realized as separate Protégé tabs: the MOBIXPL tab emulates a mobile terminal and features core visualization concepts of the original ONTOTRACK system (see Section 3) while additional tabs provide access to service ontologies, scenario data, user preference and service discovery. In the following we discuss main features and benefits of the current implementation.

5.1 Individual user views

Figure 6 shows a MOBIONT screenshot with MOBIXPL as the active Protégé tab. In this example, inside the emulated mobile device, a user-specific view to the ontology was created that comprises different view panes pointing to the base concepts Format, Program and Region of the underlying service ontology. The idea of such view panes is to point users to relevant sub-ontologies which describe certain aspects of the full service ontology. In MOBIONT individual views are stored as metadata in terms of typical usage patterns that can be activated based on the user's profile. Technically, the view metadata is coded in a separate OWL file, where concepts of scenario ontologies are treated as instances of predefined metadata classes.

While the device emulation inside the MOBIXPL tab is used for browsing the ontology and selecting preferences, the remaining application switches and panes are used to control the MOBIONT toolbox. The 'StrategySelection' tab is used to choose and to define different discovery strategies while buttons in the lower part of the MOBIXPL pane trigger the preference building and service discovery, respectively. In the current implementation, discovery strategy cannot be directly changed inside the emulated client device (by the user). However, inside the overall the MOBIONT framework the developer is free to define new strategies and add them to the system for evaluation.

5.2 Preference Selection

Similar to ontology visualization in ONTOTRACK each individual view can be gradually expanded by the user. Ontology branches which are still contracted are symbolized by triangles or as miniature sub-branches (depending on the number of descendants). For example, Figure 6 shows the sub-ontology of our radio Program formats after expansion of the top-level concepts Education, Entertainment, Information and Infotainment. The concept Entertainment itself is further expanded by one level. With respect to limited space of the mobile display, views are automatically re-centered.

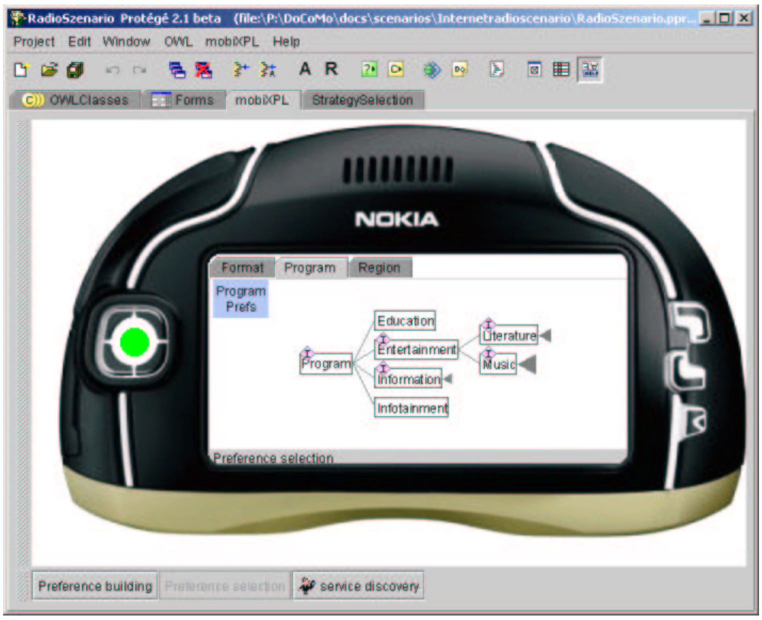

Fig. 6. MOBIXPL the graphical frontend to MOBIONT.

As motivated earlier, in MOBIONT service requests are formulated as preference expressions that are subsequently matched against available service offerings. On the client side, preference specification consists of two phases: preference selection and preference building. During ontology exploration concepts considered relevant by the user are selected for later reference during preference building. This is done by simply dragging and dropping these concepts onto the mini pane in the upper-left corner of the actual MOBIXPL pane, e.g. in the screenshot of Figure 6 concepts of the Program view can be dropped onto the 'Program Prefs' mini pane.

5.3 Preference Building

After switching to preference building mode, the MOBIXPL client displays preference tabs for each single view pane that a user has used to pre-select relevant concepts during preference selection. A preference building example is shown in Figure 7. Here, a user has identified America, Japan and Europe as key concepts from Region (a Program's origin). By default these pre-selected concepts are unbiased, i.e. considered an equally good first choice if matched during service discovery (cf. left-hand side of Figure 7).

The user is now free to define an explicit preference relation among her key concepts. In the example, the user decides to explicitly rank European programs over those from Japan or America (cf. right-hand side of Figure 7). In MOBIXPL such a preference relation is established by means of few point-n-clicks only while the preference graph is automatically re-drawn to reflect the introduced order. Besides ordering a set of key

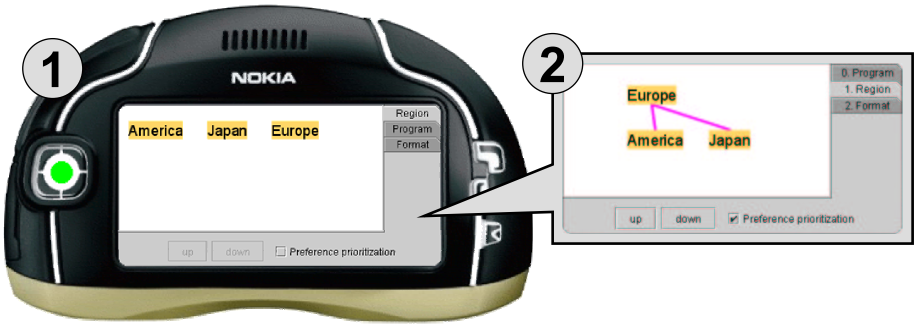

Fig. 7. Preference Building: from unbiased Pareto accumulation to biased prioritization.

concepts according to her preferences, a user can choose between Pareto accumulation or preference prioritization as the global preference aggregation scheme during service discovery (see Section 4). If preference prioritization is chosen the ordering of single preference tabs from top to bottom reflects the prioritization order. This order can still be modified via the preference building pane's up- and down-button, respectively.

5.4 Service Discovery

User preferences constructed during preference building define a service request that ultimately needs to be mapped to the underlying service ontology. MOBIONT therefore implements a flexible service discovery algorithm that can be extended through different discovery strategies. The goal of service discovery is to retrieve those service instances from the ontology that represent the best match to given preferences. The search starts at the most specific concepts of the preference relations. If no match can be found, it is gradually expanded (relaxed) to next-best matches. To this end, the aggregated preference relation (cf. Section 4) is consulted to determine a relaxation path according to the user's preferences.

According to [1], in addition to user-determined relaxation path the service ontology itself can be used for query relaxation: if no matching service instance is directly found in the concept(s) under consideration, the discovery might be relaxed to the next general concept(s). In that sense, although matching a service in the directly described concept(s) is clearly considered the best fit, a near match might still lie within the somewhat broader subsuming concept(s). To implement this discovery option, parts of the original service discovery are treated as additional preference relations with the subsumption relation as an inverse preference, e.g. if "A subsumes B" then "B is preferred over A".

MOBIONT offers support for discovery strategies that resolve how user-defined preferences are used together with "default preferences" from the service ontology. Currently, two main strategies exist that allow for a combination of both parts under Pareto- and Prioritization semantics. Note that MOBIONT is implemented in an open approach to support the definition and deployment of additional service discovery strategies. Furthermore, a build-in graphical editor allows selecting different plugged-in strategies. What is more, chain of strategies can be defined to specify which strategy (if any) should be used if a strategy fails during discovery. To allow further flexibility, each relaxation strategy can specify an own sub-editor to define and select strategy-specific features to build composed strategies.

The result set of discovered Web services will be presented to the user via a choose list with two kinds of items in order to distinguish perfect matches from relaxation matches. More concrete, if no relaxation has to be performed to find a Web service, we consider it as a perfect match. In contrast, a relaxation match is the result of some some relaxation steps (e.g. ontology-based relaxation) for which MOBIXPL will provide a graphical explanation. In such a case, the concepts along the corresponding relaxation path will be shown with additional preference edges to indicate the relaxation steps.

5.5 Metadata and Usage Patterns

As already mentioned, in addition to a service ontology which classifies related services (e.g. services offering radio streams), metadata provides configureable information about relaxation strategies for the service discovery component MOBIONT.

Furthermore, metadata is also used for user profiles. Technically, this kind of information is based on a user-profile ontology described in the OWL language format. For instance, in our current implementation users can be categorized to be novice, experienced or expert users. First, the initial individual user views are generated upon usage pattern which can be defined for each user profile. In our running example, a novice user will only see the conceptual view `Program`. In contrast, the views `Region` and `Format` are also presented to an expert user wanting to build preferences on regional information as well as on streaming formats. Furthermore, usage patterns can be used to define a "concept of interest" in a conceptual view. MOBIONT will initally expand the path from the root concept to the "concept of interest" (if any) in each conceptual view. This allows a fast and easy way to select concepts without explicitly expand all concepts along the path and event does not presuppose an overall understanding of the whole subtaxonomy. Assuming a teenage user who is mostly interested in music, a usage pattern may state that the starting concept is `Music`. Note, that this does not imply that only `Music` with all subconcepts are shown, but that `Music` is a highlighted concept shown in the center of the diplay area. Additionally, usage pattern can describe default preferences which will be automatically added to the preferences in the Preference Building mode and are used to build queries.

6 Summary and Outlook

User-friendliness will be paramount to make mobile and ubiquitous computing a commercial success allowing different types of users to access various kinds of information, anywhere and anytime. In this paper we have presented our work on a toolbox for evaluation of user interface techniques and query mechanisms for service discovery mobile devices. The implementation builds on our previous work on preference-based Web service discovery [1] and selection [2] enhanced through ontology views and search strategies [3]. In addition to core discovery functionality implemented in MOBIONT, the corresponding user interface MOBIXPL reuses graphical representation and navigation techniques of our existing ontology authoring tool ONTOTRACK [8]. We currently implement this toolbox as a plugin for Protégé while using the Protégé OWL plugin in parallel. The toolbox under development is designed to serve as an open framework with well defined interfaces for straightforward integration and composition of alternative discovery mechanisms.

The current implementation status of our toolbox is that of an early prototype. As a consequence, we have not been able to work with large-scale scenarios or to benchmark many different discovery strategies yet. Besides finishing our toolbox, we plan to extend the system towards enhanced personalization and user modelling. Currently, usage patterns are merely associated with predefined categories like novice or expert user. However, our descriptive definition of profiles in OWL allows us to extend the implemented profiling paradigm to support individual users with individual default preferences. These generic preferences could be learnt from closely observing user interactions in MOBIXPL or be explicitly defined by the individual user.

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. J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. Technical Report SMI-2002-0943, Stanford University, 2002.
6. W. Kiessling. Foundations of Preferences in Database Systems. In *Proc. of the 29th Int. Conf. on Very Large Data Bases (VLDB)*, pages 311–322, Hong Kong, China, 2002.
7. T. Liebig and O. Noppens. OntoTrack: Fast Browsing and Easy Editing of Large Ontologies. In *Proc. of the 2nd Int. Workshop on Evaluation of Ontology-based Tools (EON), in conjunction with ISWC03*, Sanibel Island, FL, USA, October 2003.
8. T. Liebig and O. Noppens. ONTOTRACK: Combining Browsing and Editing with Reasoning and Explaining for OWL Lite Ontologies. In *Proc. of the 3rd Int. Semantic Web Conf. (ISWC)*, Hiroshima, Japan, Novemer 2004. to appear.
9. Radio Locator. http://www.radio-locator.com/.
10. NullSoft SHOUTcast. http://www.shoutcast.com.