A collaborative and feature-based approach to Context-Sensitive Service Discovery

Julia Kuck
Institute of Computer Science III
University of Bonn
Germany
kuck@iai.uni-bonn.de

Frank Reichartz
Institute of Computer Science III
University of Bonn
Germany
reichart@iai.uni-bonn.de

ABSTRACT

The emergence of mobile devices has enforced the rapid evolvement of the mobile Web. We envision a new and cheap class of mobile devices, which uses Web Services to provide their functionality. These devices will be especially useful in developing countries where access to computers is difficult due to high costs or missing infrastructure. To help users finding the services appropriate to their unique needs, a discovery model is needed which takes context information and user feedback into account. This paper introduces a novel Web Service discovery approach, built up on a feature-based service context model. Our architecture also allows automated and implicit collaboration of users, who share common contexts, interests and service needs. This is especially valuable for novice users, e. g. from developing countries. So far, Web Service discovery has had little attention in the mobile domain and existing systems for Web Service discovery generally do not consider context and community oriented user collaboration. Motivated by this, our current research with the Deutsche Telekom Laboratories focuses on context-sensitive intelligence (CSI)[1] and its application to mobile service discovery.

Keywords

Context-Sensitivity, Service Discovery, Mobile Web, Web Services, Information Retrieval, Mobile Computing.

1. INTRODUCTION

Today more than 57% of the U.S. population own a cellular telephone, and even from June to December 2004, the number of wireless consumers in the U.S. has grown by over 2 million per month [7]. In the industrialized countries, mobile computing devices – like laptops, PDAs, and mobile phones – establish more and more. These novel technologies could benefit even more the developing world. Mobile devices have various advantages over conventional computers. They need much less electric current, in case of solar power no electric infrastructure is needed at all. They are cheaper in acquisition and maintenance, easy and intuitively to use, and the need of technical support especially for novice users is much lower. Further, these devices can easily be shared among a group of people, e. g. one device per village could be a first step. While the computational power and memory resources of mobile devices are low, Web Services are a promising technology to overcome this deficit. Computations are executed centralized by the service provider, improving the functionality for the everyday information access process. One very special characteristic of mobile devices is their situational embeddedness. They are increasingly capable of perceiving their current usage context. Therefore our vision is, that mobile computational devices will behave appropriate for a user's current situation, in industrialized countries as well as in developing countries. The intentions of mobile users are typically more immediate and goal-directed than the interests of users of fixed or desktop devices. Mobile users often aim at finding out specific, context relevant information [16]. Lengthy documents or browsing are typically of less interest due to devices' ergonomics. However a more appropriate mobile access to information is often still unavailable. Context-sensitive service discovery should be able to satisfy the user's specific immediate needs. Additionally, community support is of importance to enable communication and an improved exchange of information among mobile users. Users who share the same context, interests, or service needs should be enabled to constitute and join virtual communities. Community specific services and communication between their members would improve the mobile service retrieval. Users in the developing countries could collaborate and benefit from tailored discovery results for their individual needs. In this paper, we introduce an architecture for the discovery process that benefits from a sophisticated service context model. We present novel techniques for dynamic adaptation of user and service profiles to contexts, and techniques for context-aware user collaboration. This paper focuses on our goals, approaches, and contributions towards context-sensitive service discovery for the mobile Web, especially useful for the evolving technology needs in the developing world.

[1]sam.iai.uni-bonn.de/csi

WWW2007, May 8–12, 2007, Banff, Canada.
.

2. RELATED WORK

Service discovery from mobile devices differs from traditional service discovery. Enhancing existing Web Service discovery mechanisms with respect to the mobile context has had little attention in the recent years. We introduce service discovery mechanisms that have been developed in former work, involving different contextual information.

An early approach for context-sensitive service discovery in pervasive environments is gathering contextual information of users. For this purpose, the CB-Sec framework [13] includes a mathematical context model that contains different kinds of contexts (static and dynamic), representing a user. Additionally, it contains a limited service model containing the service interface, the location, and constraints regarding parameters. CB-Sec combines context-aware computing with agent technology and enables a context-aware service composition.

Another approach to context-awareness for service discovery of mobile clients is given in [10], where service authors have to provide appropriate *context attributes* for their services. These attributes are required to implement a predefined java interface. Based on these attributes, the discovery system builds a service description for the usage within the discovery processes.

Doulkeridis et al. [4] identify context for service discovery for mobile computing as *service context* and *user context.* The user context is matched against the service context during service discovery. Therefore a user is enabled to define keyword-based queries. For each query, a separate query representing the user's current context is automatically generated, and the discovery system processes these two queries. Service descriptions are rather static and the context information is used to implement a filter mechanism on common context-independent search results.

F. Klan presents in [8] a general idea of a framework enabling context-aware service discovery, selection and usage for mobile devices. Service requests are automatically extended and refined with context information. Services within this approach are characterized by the semantic service description language "DIANE Service Description". In future work they plan to examine, how *service composition* could be utilized for this context-sensitive service discovery.

A. R. El-Sayed et al. propose in [5] a context-aware service discovery architecture for pervasive computing. Their pervasive-computing environment is based on the open communications technology Jabber/XMPP[2], a distributed and *XML-based instant-messaging system.* Current situations of service providers, users, and all services within the environment are internally ontology-based represented and maintained. Based on this, a semantic matchmaking algorithm discovers the most suitable services, which become refined, filtered and ranked.

In the area of mobile service discovery, [2] present a prototype for context-sensitive discovery based on the concept of Personal Area Network (PAN) forming a network of personal devices which have a *short geographical distance.* The prototype is a hierarchical system, consisting of entities called User-nodes, Super-nodes, and Root-nodes, that perform a distributed service discovery. Users have the role of service consumers as well as service providers. The discovery process is based on matchings between user and service context.

At last, *Universal Description, Discovery, and Integration* (UDDI) as standard interoperable platform for service discovery has to be mentioned. UDDI enables companies and applications to publish and discover Web Services over the WWW. It neither provides search capabilities based on attributes nor offers a suitable mechanism to personalize or rank the query results. Because of these limitations, UDDI could only be used as addition to a context-sensitive approach, but on its own it does not challenge the task of discovering mobile Web Services in a dynamic environment.

[2]www.xmpp.org/

3. ARCHITECTURE

One important aspect in terms of *Context-Sensitive Intelligence* (CSI) [11, 12] concerns the detection and selection of adequate services. Service discovery will deliver an *ordered selection* of services with regard to the user's service needs. If there is a great amount of possible services, a user does not have time or does not wish to spend time on dealing with a bulk of services. The key goal of our service discovery architecture is to retrieve services, which might be useful or relevant to the user, and which are ranked by the relevance for the individual user's special need. Because mobile users have situational needs, we propose novel techniques considering context information to improve service discovery. This chapter describes the architectural design of the *context-sensitive service discovery system* (CSDS) [9]. The configuration of processing nodes and components of the envisioned CSDS is shown in figure 1.

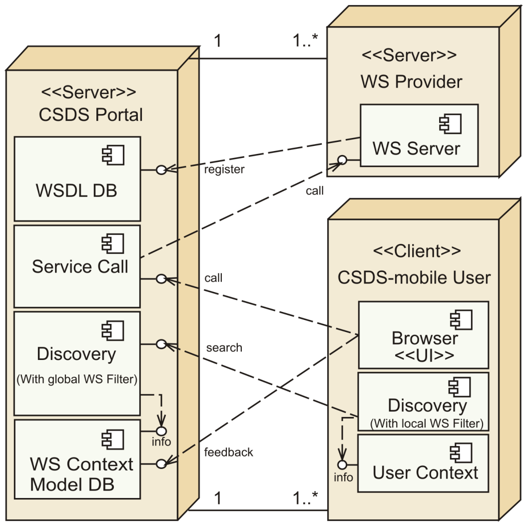

Figure 1: Deployment Diagram of CSDS

The following discussion focuses on: (i) Context; (ii) Formalized Service Discovery Model; (iii) Composition search and service composition. To assure a generally applicable discovery system, we retain established standards of the Web Service architecture platform. For this, we only require WSDL service descriptions of service providers or UDDI-repositories as basis for CSCS service descriptions. The second requirement is context information, delivered by personal and environmental sensors.

3.1 Context

The context-sensitivity for mobile service discovery that we promote consists of the *user context* and a *service context model.* Based on this classification, matchings between user and service contexts could be found, facilitating a sophisticated mobile service discovery.

User Context: As illustrated in figure 2, a user context comprises three categories. The first category describes the personal background of the mobile user. This are user data about gender, age, domicil etc. The environment category provides situational context information like the actual location and the environment. The third category captures the user's "information world" – e. g. read documents and visited Web pages – thus reflecting the user's interests. Owing to computational reasons, there is no voluminous user context model on the mobile client. Only the essential user data reside there.

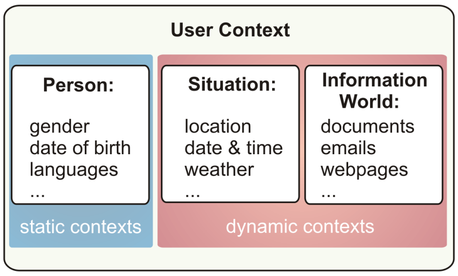

Figure 2: The context of a mobile user

Service Context Model: We introduce the service context model, depicted in figure 3. For every Web Service we detect its contextual information. We distinguish between two automatically derived categories. The first category is called "WSDL". While there exist several approaches for the description of Web Services, the *Web Services Description Language* (WSDL) is the primary description document of Web Services. Generally a service provider delivers the WSDL document to describe an offered service. A WSDL document defines the methods of a Web Service, their parameters and results, and a service's region of interest (location). Such a document enables clients to use the methods of the Web Service in a correct manner. While WSDL is commenly used to receive syntactic information of how to call a Web Service, we additionally extract semantic information. By an automated analysis different kinds of semantic data can be derived from WSDL-documents [14]. For example it is possible to determine the meaning of:

- parameter names and operation names,
- textual documentation of the entire Web Service,
- textual documentation of operations.

The second category of the service context model is called "Feedback", and contains user-based information like the service's popularity. Therefore, this category consists for example of:

- the user context at the time a recommendation is given,
- the query it was judged relevant to,
- the acceptance of the service.

We assume, that most people recommend a service shortly after invoking it. Our service context model needs to represent all of this information in a unified way. The different kinds of data within the WSDL- and the feedback-category of the service context model are called *features*. The extractable features of a Web Service are manifold. Hence this model has to offer the option to store a vast number of previously unknown features. The quality of the discovery process can then be enhanced by using these additional special information stored in an instance of the service context model. Such a model could be specified in RDF[3] enhanced by an ontology described in OWL.[4] Through the semantic web movement, several approaches to deal with automatically extracted meta data have been developed and successfully tested [17]. The involvement of these techniques in further research has high potential to improve our feature extraction procedure.

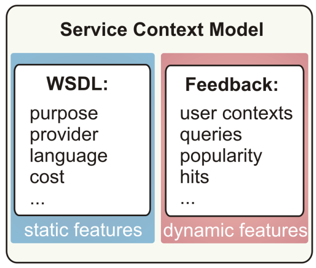

Figure 3: Sample features of a Web Service's context model

While existing architectures for service discovery tend to assume a static world, with never changing service descriptions, recent research has highlighted that adequate service descriptions as well as user representations can dynamically change [3]. In the CSDS we also distinguish between *static* and *dynamic* contexts and features, see figures 2 and 3. Dynamic user context is useful if changing user context represents changing service needs. By means of the dynamic features it is possible to improve the modelling of a service by using user specific data concerning the service. This acts as implicit or explicit relevance feedback by the system users. In general our dynamic approach should facilitate a matchmaking algorithm for a high grade context-sensitive service discovery.

3.2 Formalized Service Discovery Model

With regard to the user context and the context service model an adequate service discovery system should find the right services when they are needed, either on demand or when new services are provided. This is like an integrated automated information push and pull cycle. Because of the similarities between the challenges of information retrieval and service discovery, the adaptation and advancement of established information retrieval techniques for the discovery of services is a promising step.

Representation of Service Needs: To describe the user service needs, a user context is needed that is automatically generated by an application on each discovery request. In order for it to work, at least one kind of user context has to be provided to the service discovery model. Without any context, there would be no need for context sensitive discovery. A service need n of a user u can be defined by the pair $n_u := (q_u, c_u)$. Here the query is for simplicity represented by the query terms q_u. Users can always formulate their actual service need via keywords. c_u denotes the user context. For presentation issues only one kind of context is specified; this could be for example the city in which the user currently resides. This formalism could easily be extended by further context categories.

[3] www.w3.org/RDF
[4] www.w3.org/TR/owl-features

Representation of Services: As stated before, a successful service discovery demands a detailed service context model, describing the static and dynamic features. The services $\mathbb{S}$ which are known to the system are represented by their extracted features according to the service context modell. So a new set $\mathcal{S}$ of features is generated based on a feature-extractor applied to each service of $\mathbb{S}$. In the following we only use a minimal model.

Minimal Service Context Model: A minimal service context model contains at least the following features:

1. the purpose $\mathcal{P}$ of the service,
2. the associated query terms $\mathcal{Q}$ of the users,
3. the language L of the service.

The first feature $\mathcal{P}$ is generated as follows: We split parameter names and operation names as well as the documentation terms of the available WSDL-documentations into their constituents. E. g. we split 'getCanadaWeather' in 'get', 'canada' and 'weather'. Subsequently we normalize the terms of the splitting as well as terms from the documentations by stemming algorithms. Additionally, we use a stopword list to filter out non-specific terms like 'get'. The resulting multiset of terms defines the first feature, representing the purpose of the service. Within the second feature $\mathcal{Q}$, query terms of users are stored. These are the queries users have submitted to find this service. The third feature of a service L represents its language.

In former work our research group has shown that standard information retrieval techniques can be used for service discovery [9]. There it was demonstrated that a modified BM25 formula can be used for service discovery based on contexts. We propose that by means of the use of a converting function our new context service model can be directly incorporated in our former approach.

Adaptation to former Approach: In our former model the user context was modelled by a user profile containing the user's *personal terms* $\mathcal{PT}_u$. Our user context conversion function C_U maps the context c_u to the collection $\mathcal{PT}_u$ that represents (describes) the context. In the above example where the context consists only of a city it would be the identity $\mathcal{PT}_u = C_U(c_u) = c_u$ hence $C_U(c_u) := id$ for all values of these kinds of context. For the service context model, a general class of functions for the deduction of terms from features is being developed and improved. These functions map the static features of a service s to the multiset $\mathcal{SF}_s$ and the dynamic user generated features to the multiset $\mathcal{UF}_s$. In general the following approach maps the features of service s to the multiset $\mathcal{F}_s$. The underlying set of elements of the multiset are the *service terms* $\mathcal{ST}_s$, representing the service context model after some form of term extraction through an extraction function. The multiplicity function m for the multiset is used to formalize how much each of these terms represents the semantic of the service. These functions are a central piece of our context-sensitive discovery model. Further research has to be conducted on which kind of functions are applicable.

In the following, we present sample functions for our minimal service context model. The purpose feature $\mathcal{P}$ can be mapped to the multiset $\mathcal{SF} := (\mathcal{P}, 1_{\mathcal{P}})$ where the multiplicity function is the characteristic function $1_{\mathcal{P}}$ of the set $\mathcal{P}$. The query terms $\mathcal{Q}_u$, which have been associated (via feedback by the user u) with the service, are joined over all users to $\mathcal{UF} := \biguplus_{u \in \mathcal{U}} \mathcal{Q}_u$ which creates the *user features*. Therefore the multiset of the features $\mathcal{F}$ of the service could be defined as follows: $\mathcal{F} := \mathcal{UF} \biguplus \mathcal{SF}$. The underlying set of elements of $\mathcal{F}$ are the *service terms* $\mathcal{ST}$ of the service.

To generate a *ranking* of suitable services we can generate the final weight of a service s – based on the personal terms $\mathcal{PT}$ and the service terms $\mathcal{ST}$ – with an extended modification of our Okapi BM25 formula adaptation described and motivated in Kuck et al. [9]:

$$w(s,q,u) = \sum_{t \in (q \cup \mathcal{PT}_u)} w_t^{(1)} \cdot \frac{2tf}{dl/avdl + tf} \cdot \frac{2qtf}{1 + qtf}$$

where tf is the term frequency of term t, qtf is the query term frequency of term t in query q, dl is the cardinality of $\mathcal{F}_s$ and $avdl$ is the average cardinality of all service descriptions $\mathcal{F}_s$. Our adaptation of the Robertson-Sparck Jones weight [15] delivers $w_t^{(1)}$ (detailed information of this adaptation is given in [9]). $w_t^{(1)}$ describes the inverse collection frequency endorsed by relevance feedback information. The language feature L is used afterwards to filter the retrieved results according to the language of the query terms. Extensive service context models could deliver language specific features in a more comprehensive way.

It's of special interest that the dynamic features will be the part of our model where the implicit cooperation of the users is fostered. Through this, a system is established which creates a user driven information generating process. In the spirit of the Web 2.0 a user driven dynamic content – in contrast to the broker given static WSDL descriptions – is founded. This could be interpreted as a document-expansion. Additionally, our research group is currently developing adequate context-sensitive techniques to detect groups of users with similar profiles. Our plan is to develop further mechanisms which allow automated service recommendations between single community members.

3.3 Composition Search and Service Composition

Context-sensitive search should not only deliver single Web Services, but also service compositions to achieve the desired functionality. For example, if a Web Service operation satisfying a search requirement is not available, it is valuable to return a composition of services that satisfies the requirement in combination. Additionally, users should be enabled to build and publish their own service compositions.

Composition Search: Composition search should not only deliver single services, but also suggestions of simple service compositions. This could be managed by comparing the output operations of one service with the input operations of another service. Service descriptions with high similarities could then be unified according to their matching semantics, and a context-sensitive ranking via our BM25-adaption could be generated. This ranking would then weight the service combinations. A detailed approach towards composition search for CSDS is a promising next step and will be elaborated in future work.

Service Composition: The *Service-Oriented Architecture* (SOA) proposes a new software architectural paradigm that is based on loosely coupled software components, which are deployed and located across the WWW. The W3C Glossary defines SOA as "a set of components which can be invoked, and whose interface descriptions can be published and discovered". These components offer interfaces satisfying the services in a service-oriented architecture. Web service technology is a base for the implementation of SOA. More and more application scenarios aspire at re-designing IT during the *use time* and in the context of use [18, 1]. For context-sensitive service composition, we propose an approach to support end-users while tailoring their applications themselves. Our cooperative tailoring approach offers service consumers three different operations: 1) Adding new service applications within the workspace; 2) Linking two services according to their interfaces; 3) Publish/recommend the linking of services to other users. Tailoring mechanisms should be provided at different levels of complexity [18]. In the area of mobile context-sensitive service discovery, most end-users are non experts in Web Service technologies and in composition languages like the *Business Process Execution Language* (BPEL). Therefore a simple mechanism for service composition should also be offered for the novice users. Regarding the works of [6], the development of a simple programming language for end-users that could be automatically transformed into BPEL would be a promising next step.

4. CONCLUSION

In this paper, we promote a novel approach for context-sensitive service discovery in the mobile Web. This allows escpecially users relying on devices with limited ressources access to services their devices are not capable to provide. The provision of computing power by service providers is for example usefull for business users who now can use services to analyze important business data supporting their decision processes. But also users from developing countries could benefit from this to overcome the lack of access to information services. Therefore our approach contributes to cross-national sharing of computing power and information. The mobile service discovery that we present is built on a feature-based *service context model* and the *user context*. The service context model enables the integration and easy extension of numerous features representing the manifold kinds of service contexts due to the application scenarios for the mobile Web. The research contributions of our work arise from (i) context-based document expansion and result filtering due to various static and dynamic service features; (ii) query expansion with user context; (iii) facilitated collaborations and recommendations between mobile service consumers; and (iv) a motivating proposal for mobile Web access combined with novel Web Service technologies for the developing world. The extensive processing of WSDL leads to an improved discovery, and benefits the service providers since they can keep on publishing their services in this common way. In order to achieve a high discovery quality, we proposed an adaptation to a well-known information retrieval ranking function. Additionally, we have done first steps towards a context-sensitive composition search as well as a user-tailored service composition. In the future, we will improve our architecture by investigating further service features and techniques enabling non-term based filtering for the ranking. Our future plans also focus on an improved treatment of the implicit and explicit recommendations to develop a community model, which is new for Web Service discovery. Privacy issues and an appropriate classification of client and server side computation and data handling will be addressed in the future.

5. REFERENCES

[1] S. Alda. Component-based adaptation methods for service-oriented peer-to-peer architectures. *Ph.D. Thesis of the University Bonn*. Germany, 2006.

[2] R. Balken, J. Haukrogh, J. L. Jensen, M. N. Jensen, L. J. Roost, P. N. Toft, R. L. Olsen, and H.-P. Schwefel. Context-sensitive service discovery experimental prototype and evaluation. *Wirel. Pers. Commun.*, 40(3):417–431, 2007.

[3] S. Cuddy, M. Katchabaw, and H. Lutfiyya. Context-aware service selection based on dynamic and static service attributes. In *IEEE Int'l Conf. on WiMob*, pages 13– 20 Vol. 4, 2005.

[4] C. Doulkeridis, N. Loutas, and M. Vazirgiannis. A system architecture for context-aware service discovery. *ENTCS*, 146(1):101–116, 2006.

[5] A. R. El-Sayed and J. P. Black. Semantic-based context-aware service discovery in pervasive-computing environments. In *Proc. of the 1st WS on SIPE at IEEE ICPS*, 2006.

[6] I. Gavran, A. Milanovic, and S. Srbljic. End-user programming language for service-oriented integration. In *proc. of WDAS*, CA, USA, 2006.

[7] M. Kamvar and S. Baluja. A large scale study of wireless search behavior: Google mobile search. In *proc. of the Int. WWW Conf.*, Edinburgh, UK, 2006.

[8] F. Klan. Context-aware service discovery, selection and usage. In S. Brass and A. Hinneburg, editors, *Grundlagen von Datenbanken*, pages 95–99. Institute of Computer Science, Martin-Luther-University, 2006.

[9] J. Kuck and M. Gnasa. Context-sensitive service discovery meets information retrieval. In *proc. of 5th IEEE Int. Conf. on Pervasive Computing and Communications*, New York, USA, 2007.

[10] C. Lee and S. Helal. Context attributes: An approach to enable context-awareness for service discovery. In *Proc. of SAINT*, pages 22–30, 2003.

[11] H. Mügge, T. Rho, D. Speicher, J. Kuck, and A. B. Cremers. Towards an infrastructure for Context-Sensitive Intelligence. In *SAKS WS*, Kassel, Germany, 2006.

[12] H. Mügge, T. Rho, M. Winandy, M. Won, A. B. Cremers, P. Costanza, and R. Englert. Towards context-sensitive intelligence, EWSA. 2005.

[13] S. K. Mostéfaoui and B. Hirsbrunner. Towards a context-based service composition framework. In *Proc. of ICWS*, pages 42–45, 2003.

[14] U. Radetzki, U. Leser, S. C. Schulze-Rauschenbach, J. Zimmermann, J. Lüssem, T. Bode, and A. B. Cremers. Adapters, shims, and glue - service interoperability for in silico experiments. In *Bioinformatics, Advanced Access*. OUP, 2006.

[15] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *JASIS*, 27:129–146, 1976.

[16] W3C. Mobile web best practices 1.0 - basic guidelines, 2006.

[17] C. A. Welty and J. W. Murdock. Towards knowledge acquisition from information extraction. In *proc. of ISWC*, pages 709–722, 2006.

[18] V. Wulf, V. Pipek, and M. Won. Component-based tailorability: Towards highly flexible software applications. In *IJHCS*, 2006.