SERVICE PERSONALIZATION FOR USER SUPPORT

I. Vaynerman
Heinz-Nixdorf Endowed Chair for Practical Informatics
Institute of Computer Science
Friedrich-Schiller-University Jena
Jena, 07743, Germany

ABSTRACT

Mobile users need access to information sources. However, for a number of reasons, the traditional representation of query results is not useful in mobile environments. There is also a pressing question of using network services to provide mobile users with required information. The potential to dynamically bind services at run time is one of the big advantages of service oriented computing. However, dynamic service binding requires rather sophisticated service requests. These requests need to be precise enough to allow for automatic invocation of the service without user intervention, while at the same time flexible enough to allow finding all possible candidate services and could be rather clearly represented on a mobile device. Obviously, it is not trivial to formulate such requests. In this paper, we examine how the user could be supported in this complex task.

Keywords: service, personalization, priorities, mobile device, user support

INTRODUCTION

To provide mobile users with sophisticated means of information access is a problem of growing importance, in particular since the numbers of mobile users are growing continuously. Although mobile devices and wireless networks are steadily improving, some restrictions will remain for the foreseeable future. As compared with their stationary counterparts, mobile devices have smaller displays and less comfortable input possibilities.

Nowadays, the problems of mobile device limitations are solved either by splitting the information volume into smaller units or by picking the most important results. These existing approaches assume that a user’s required information can be recognized beforehand. Moreover, the users themselves are believed to be able to determine and precisely formulate which information they need. In our opinion, both of these assumptions are unrealistic. It is thus necessary to develop new specific information delivery procedures for mobile and wireless users to adapt the information to the characteristics of these systems and user demands. Such a way of information delivering could be the personalisation of network services. In the next sections we present our approach of service personalisation.

SERVICE ORIENTED COMPUTING

Service orientation is a promising paradigm for distributed resource usage. Service orientation has the potential to fully automate resource sharing. Full automation makes particular sense for repetitive queries. As an example we will consider several steps of a business process on some mobile device (Figure 1). Let us assume that we should create a report in PDF format, print it and then send it via a courier service to our headquarters. We are using some mobile device e.g. a PDA or smartphone. At first, we create a report in .RTF format (e.g. in Pocket Word), but to make the further operations we need a .PDF document. The process of converting into PDF format needs high processor performance and the execution on a mobile device can take very long. Therefore, it makes sense to use a suitable network service instead of using the mobile application. Thus, we have to find and call such a service. So, after report converting the user needs to print it. He can do this in two ways. The first one is to use a rigidly determined printer. The user knows the name of this printer e.g. from previously using it, and sends the report to it. If our business process makes this step automatically we need to have 100% guarantees that the report will be printed.

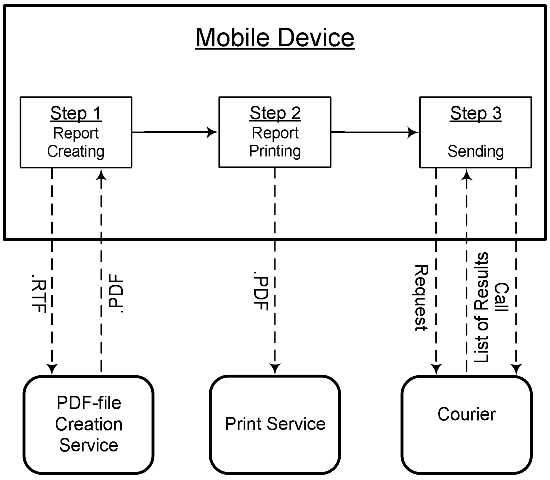

Figure 1. Business Process on a Mobile Device.

However, it could be that the user has gone from one place to another (i.e. from one building to another) and there is no connection with the remote printer, or it doesn't work. In this case, we should be able to choose the suitable printer each time before printing. Thus, it arises the second way to print the report: using a network printing service. Instead of calling the remote printer, we should send the request to print the report with some parameters to the service. Possible parameters would be, e.g., resolution, time or place, where the user can pick up the printed report. In an ideal case, the user should only send a request; the service chooses the appropriate printer and takes all necessary actions. When the report is printed and picked up, we need to send it by courier. Again, a service could be used. Obviously, if these three steps are carried out quite frequently, it should not require human intervention each time. Therefore, what is needed is a system that allows automatic service matching and binding. With such a system it would be possible to describe once which functionality should be provided and then automatically find the best match for this request at each invocation.

Full automation of service matching and binding requires complex service requests. The reason for this is that in conventional systems, the search results are presented to the user who then filters them and selects the most appropriate service provider. In a fully automated system, this manual filtering and selecting does not take place. This means that the initial request formulated by the user needs to be precise enough to strictly restrict the search result to services that offer exactly the functionality needed. Also, the request needs to contain all the information that is needed to rank among possible service providers. At the same time, service offers need to be described precisely, too, so that automatic matching becomes feasible. Within the DIANE project [3], suitable language for service request and offer description, DIANE Service Description (DSD), was developed [1]. It allows flexibly describing and processing the semantics of service offers and requests. If offers and requests are formulated in this language, automatic service matching and binding is possible.

Unfortunately, user requests become rather complex and nontrivial, if they are to match the requirements mentioned above. In our opinion, it is unlikely, that the user will succeed in formulating such a request without support by the system. The most likely scenario is that the user will formulate an initial request and will then check with a trial run whether the services found with this request match his expectations. If this is not the case, the user will adapt the request, rerun it, check the results once more and so on until, finally, the desired result set is retrieved. The request that produced this result set is then stored and used in future iterations of the process.

REQUIREMENTS FOR USER SUPPORT

Let us return to the third step of the business process introduced above. When the report is printed and picked up, we need to send it by courier. The main problem is to choose the appropriate courier company. Again, the user should be able to specify once what the characteristics of such a company are; it should then be ensured that during future iterations of the business process the request is processed and the optimal courier company for the task at hand chosen. The request can contain a lot of parameters and priorities, since the decision on what courier service to use may be a complex one. The role of parameters in such a request could play: Price of Delivery, Term of Delivery and additional courier services (Online Billing Electronic Shipping, Delivery Tracking). Priorities here are the user-specific preferences in request parameters. Other words some request parameters could be more important to user than others. In DSD as such a mechanism to express user priorities in request serves “Connecting Strategy” (CS). Basically, CS is some mathematical formula, which integrated in each element of request. It specifies how different request parameters, which concern to this very request element, should be together combined. As a little example we consider the possible CS for courier service request: *(price*0.8 + term*0.2)/2*. Here *price* and *term* are the results of matching of service request and service offer. The coefficients *0.8* and *0.2* are actually the mathematical representation of user priorities for request parameters “Price of Delivery” and “Term of Delivery”. These coefficients show that prise in this request is much important for user than term. During the request processing all its CS’s will be evaluated in order to calculate the estimation of service offer. More details about CS and DSD you can find in the documentation of DIANE Project.

Thus, for a successful service request, i.e. one that results in the choosing of a matching service, the user needs

- To know which parameters can be specified for this particular type of service
- To determine which settings extend of each parameter is acceptable to him. For instance, what deviation in delivery price is acceptable? If user sets as value of prise parameter 10 euro, then the value 10,50 euro is still suitable to him, or not? The bounds of suitable parameter values should be defined by user in service request.
- To describe how different parameters should be prioritized, that is which parameter is how important. For instance, if the price of delivery is more critical for user than term, then he should be able to convey these priorities in the service request.

All these demands make the formulation of the query rather hard for the user. Also, if the list of the answers contains a lot of appropriate answers with similar characteristics, it should be possible to automatically rank them. Again, the parameters for the ranking need to be provided in the user query.

We mentioned above, that the user will probably not be able (even with support by the system) to come up with the “perfect” request, i.e. the request that produces exactly the intended output, from scratch. Rather, the formulation of the query will be an iterative process involving trial runs. These trial runs should be realized in a way that minimizes network load. In the next section, we take a closer look at the determination of preferences in the user request.

PRIORITY PARAMETERIZATION OF THE USER REQUESTS

In the previous Section we have shown that the process of service using becomes more and more complicated and there are a lot of objective reasons to provide user with means of support during the initial request formulation and further interaction. In our opinion the most suitable way to realize such a support is the service personalisation. Under this concept we can understand a process of service modification, which allows to consider user preferences during the interaction. So, user should be able to transmit his personal abstract interests (preferences) in some formal form that would be machine-understandable and enough flexible in dynamic iterative communication. Here we should more closely consider this term “abstract interests”. What actually are these interests? To answer this question we should determine what usually does user to define a request. User sets the concretes values of request constituents, which basically are the request parameters. Hence he could have personal preferences for these parameters. In the rest part of this article we will name these favours “priorities”. So, as we have said above, user should be able to transmit his priorities to the service, another words user priorities should be integrated in some formal way direct into the request. We name this integration “priority parameterization of the user request”. To explain this complex concept we consider the concrete example.

Let us consider using of the courier-service from the previous section more closely. The user needs to send a report and he is searching for the company, which offers the best conditions. For these purposes he uses the network searching-service. The user defines the requirements for the courier-company. Now, it is necessary to determine the relations between the request parameters. These relations define the user's priorities, i.e. the hierarchy of the parameters. The priorities show what service aspects are more important than others. In other words the priorities define what request parameters are more critical for the user at service searching. They help the user to express his preferences in a request. During the processing of the request, it is necessary to separate the appropriate service offers from the not suitable. Also, the best matching service offer has to be determined. The DIANE Service Description (DSD) includes methods and algorithms to perform the matching. However, for these methods to work, the user request needs to contain precise information. For instance, it may be necessary to specify how certain attributes of an object should be matched. That is, the user may need to tell the system when to consider two values to be similar or which value of two given values to prefer. For example, with price it is better when it is smaller, but with printer resolution this is not so. This information needs to be provided somewhere. There is also a question about the range of parameter value. If there is no exact answer, in what borders it is reasonably to search for it? Since the answers to these questions need to be provided as part of the user request, the system should guide the user in pointing out what additional information is needed to allow for a meaningful matching of the user's request to service offers.

What we have described up to now are support mechanisms that will ensure that the user request is syntactically correct and contains all the information needed to perform service matching. However, while this is a necessary first step, it does not guarantee, that the user request does indeed describe the service the user is looking for. Therefore, the system should offer a mechanism that allows the user to perform trial runs of his requests. The mechanism should then display the results obtained and allow the user to adapt the request parameters. Even better would be a system that displays the results obtained, allows the user to indicate how close they are to the intended result and which then automatically adjusts the request parameters to obtain results closer to the intended ones.

In our work we have developed three approaches to realise such a complex set of support mechanisms. These three ways are based on the methods from neuronal networks theory, fuzzy-logic and mathematic functions. We separate the different types of user request and user priorities and according to the concrete case the specific methods are used. To provide user with means of efficient service using we are developing the Semantic User Profile. There user priorities could be saved and stored in a form of ontology.

RELATED WORK

In the sections above, we have shown that user support is required to enable users to successfully formulate complex requests. The main difficulty in formulating these requests is the correct prioritizing of user preferences. Therefore, in this section, we take a look at existing approaches that allow users to formulate preference based queries. This topic has enjoyed quite a lot of attention over the last few years. We will show that the existing approaches offer a valuable basis for our work. However, we will also show that they fall short of providing the kind of user support needed here.

Skline Queries [8] are preference based queries that return as a result the "skyline" of best results. These are those results that are better than others along at least one dimension, i.e. pare to optimal. Moving along the skyline is equivalent to shifting the relative importance of the different dimensions. With respect to our problem, a skyline algorithm seems a good starting point to find a set of candidate services. However, in addition to what the skyline algorithm provides, we need a method to decide which point of the skyline to choose as the best match of the user's expectations. Unfortunately, the existing algorithm does not offer any support here.

Preference SQL [6] is more flexible and allows the combination of different preferences not only as pareto preference, but also as prioritized preference or as a weighted sum of preferences. Thus, preference SQL allows expressing a wide variety of preferences. However, Preference SQL encounters the same problem that the formulation of requests with DSD poses: It is difficult for the user to judge beforehand, how a certain setting will influence the query result. Preference SQL offers an explanation component that at least allows comprehending how a result was produced once this has been done. Again, Preference SQL seems like an interesting basis for implementation of preference based user requests. It does not address the fundamental problem described in this paper, that is how to come up with the correct request in the first place. The same is true for the XPath counterpart of Preference SQL, Preference XPath [5].

Service Globe [4] is another example of a system that allows (at least to a certain degree) the formulation of preference based queries. Again, no support for designing correct queries is given.

CONCLUSION

In this paper, we have shown that automatic service matching and binding require user support in formulating appropriate service requests. These requests will have to be preference based. We have discussed existing approaches to preference based querying and have shown that while they offer methods to compute query results based on a given query, they do not support the user in asking the correct query in the first place. Given the complexity of the queries required here, we believe that such a support is an absolute necessity. In our future work, we intend to develop a system that offers just this kind of support. This system should at the same time support the user efficiently and make economic use of sparse network resources.

The information about our previous works in the area of network services can be found in [10–13].

REFERENCES

1. Klein M., König-Ries B. "Combining Query and Preference - An Approach to Fully Automatize Dynamic Service Binding". In: *Proc. of IEEE International Conference on Web Services (ICWS 2004),* 6.-9. Juli 2004, San Diego, CA, USA.
2. Klein M., König-Ries B. "Coupled Signature and Specification Matching for Automatic Service Binding". In: *Proc. of European Conference on Web Services (ECOWS 2004)*, 27.-30. Sept. 2004, Erfurt.
3. D I A N E - Projekt http://www.ipd.uka.de/DIANE/
4. Keidl M., Seltzsam S., Kemper A. "Reliable Web Service Execution and Deployment in Dynamic Environments". *TES* 2003, pp. 104-118.
5. The official web-site of WWW Consortium http://www.w3.org/TR/xpath
6. Kießling W., Köstler G. "Preference SQL - Design, Implementation, Experiences". VLDB 2002, pp. 990-1001.
7. Semantic Web Services http://www. daml.org/ services/
8. Kossmann D., Ramsak F., Rost S. "Shooting Stars in the Sky: An Online Algorithm for Skyline Queries" In: *Proc. of VLDB 2002,* Hong Kong, China, 2002.
9. Klein M., König-Ries B. "A Process and a Tool for Creating Service Descriptions based on DAML-S B". In: *Proc. of 4th VLDB Workshop on Technologies for E-Services (TES'03)*. Berlin, 8 September 2003.
10. Koenig-Ries B., Popov D., Muelle J., Plechova O. "Multidimensional Query Result Navigation for Mobile Users". In: *Proc. "Computer Science and Information Technologies CSIT'2002"*. University of Patras, Greece, 2002,. http:// www. lar.ee.upatras.gr/ csit2002/.
11. Popov D.V., Vainerman I.A.: "A Prototype of the System to Provide Mobile Users with Services in Wireless Networks ". In: *Making Decisions under Indeterminacy Conditions*. USATU, Ufa, Russia, 2003, pp.14-22. (In Russian).
12. Yussupova N.I., König-Ries B., Popov D.V., Vainerman I.A. "Data Structures for a System to Dynamically Provide Mobile Users with Information in Wireless Networks". In: *Proc. of the 5th International Workshop Computer Science and Information Technologies CSIT'2003*. Ufa, Russia, 2003, Vol. 1, pp. 100-107.
13. Yussupova N.I., Popov D.V., Vainerman I.A., König-Ries B. "RDF-Technologies to Provide Mobile Users with Services in Wireless Networks". In: *Proc. of the 6th International Conference Complex Systems: Control and Modeling Problems*. Samara, Russia, 2004, pp. 203-208.