Elicitation of Preferences for Web Service Compositions

Carolin Michels and Sudhir Agarwal

Karlsruhe Institute of Technology
{michels,agarwal}@kit.edu

Abstract: Preference elicitation is often used in e-services to create product recommendations for their customers. We present an approach for applying preference elicitation techniques for a semi-automatic query generation system for Web service compositions. These techniques are used for identifying the user's demands on the desired Web service composition and furthermore the trade-offs between interdependent or conflicting preferences. In our iterative process the query is automatically (re-)formulated to present results to the user, so that he can detect undesired characteristics and revise his specifications until he is presented a satisfiable result.

1 Introduction

For years, online shops have provided functionalities to make their customers feel comfortable even though they cannot – as opposed to real world shops – offer advice and support in person.[1] These functionalities are attempts to personalize online shops, like for example a personalized greeting when entering a shop: If a salesperson knows us, he greets us with our name, an online shop does in most cases the same, when we log in. If a salesperson knows, which products we bought in the past, what we like in general or for which purpose we buy at his shop, he can suggest products, in case we are indecisive. These suggestions are not only helpful for the customer, but also for the salesperson as far as he sells more products and creates customer loyalty, which is why many online shops provide product recommendations, too.

In contrast, if a user looks for a Web service composition he is dependent on himself alone. Most Web service composition approaches assume that the user is capable of identifying his demands and expressing them formally in a Web service composition query. For those cases, in which he is not, we want to provide a Web service composition recommendation system similar to a product recommendation system.

There are various reasons why a user may need support for creating a Web service composition query. First of all, he actually may not know how to formulate his request or he may not be capable of identifying his concrete preferences. Also, he may be oblivious of the granularity of possible compositions, of the difference between his goal (*fundamental objectives*) and the *means* he wants to use for achieving his goal (cf. [SPM06]) or even

[1] Except for hotlines and email support, which pose a safety net, if something goes wrong and the user is not able to solve the problem with the online functionalities.

of the information needed to distinguish those services which may be suited best for his needs. Furthermore, dependencies or contradictions between his requirements may not be obvious but crucial and a system pointing them out and/or inquiring could help expressing the actual concrete preferences. Sometimes, the user is actually not aware of the fact that his needs and the possible solutions differ. A system explaining to him why and how a solution deviates from his requirements may be useful.

Secondly, a (successful) Web service composition process requires more information about the user and his requirements and preferences than an atomic Web service discovery (which is of course a part of the aforementioned): not only information about *what* the user wants (and therefore the solution) is required but also about the *how* (cf. [SPM06]). As an advantage, a Web service offers more possibilities to derive information about a user and his requirements and preferences and thus provides more means to help the user. One can think of various steps in the Web service composition process in which a preference elicitation could be helpful: (i) query generation for Web service composition, (ii) presenting the Web service compositions to the users and (iii) revising the proposed solutions. We will consider all these aspects in the process presented in Section 2.

In this paper, we present a process for preference elicitation for Web service compositions. First, in Section 2 we introduce the information used in our approach and the specifications for the preferences which have to be considered and handled for the query generation. We continue with an outline of our approach and the factors used for ranking Web services. In Section 3 we compare former research for preference elicitation. Finally, in Section 4 we give an outlook based on the proposed process model.

2 Request generation and handling

Before we describe the details of our approach, we define some fundamental terms: **Hard constraints** are requirements that must be fulfilled by the composition, i.e. the user would reject every solution that did not cover all given hard constraints. Hard constraints are fulfilled or not, e.g. the constraint "air mail delivery". In contrast to hard constraints **preferences** need not necessarily be included in the solution, but if a user is confronted with two compositions, one fulfilling a preference to a larger degree than the other, he would select the former. Preferences can be fulfilled partially and weighted by the user (e.g. "I'd rather have a fast delivery than a cheap one"). **Longterm** preferences and hard constraints are conditions that the user wants to be fulfilled every time he composes Web services, e.g. "I always prefer those services by company X". In contrast **ephemeral** preferences and hard constraints apply only temporarily (due to changed/temporal circumstances) [PFT03]. **Dependencies** can occur between services of a composition or one or more services of the composition and the overall composition (see [WS09]), i.e. those parts of a composition that interact. For example a service may be depending on the output or successful execution of another service. Additionally we regard dependencies between the preferences, if e.g. one preference (e.g. "fast shipment") is influenced by the fulfillment of another preference(e.g. "cheap shipment").

The query for Web service compositions is generated by considering three different information sources: (i) The user **profile/history**, e.g. his domain, expertise, former preferences, (ii) the **context** of the user, e.g. field of work, current environment, search context and (iii) the user's **goal definition** and further specifications stated in a form or questionnaire, e.g. input and output parameters and functional properties. While the user profile contains longterm information, i.e. general information that in the majority of the cases holds or aggregated information from previous instantiations, the goal definitions and the questionnaire comprise ephemeral preferences and constraints. The user context can cover longterm properties, e.g. the domain of the user, or properties that change from time to time, e.g. the location context of the user. We use ontologies to represent the user's expertise, his context and the services' inputs, outputs, preconditions and effects (IOPEs, cf. [SPH04]), which should include the terms used in the goal definitions. The inclusion of IOPE ontologies allows furthermore the representation of general dependencies by including a corresponding relation. Consider for example the dependencies "shipping costs depends on mode of shipment" and "total costs depends on shipping costs". Further deductions are possible with reasoning on the ontologies, e.g. if a user has a location context that induces time constraints (e.g. at the airport), he probably wants to have a fast service. Furthermore, problems arising due to the user's expertise, e.g. different wording or lack of knowledge of precise terms become treatable. The ontologies can be used for adapting the questionnaire with different wordings to the user's expertise and making suggestions for relaxing and adjusting the query (see steps 4 and 7 below), e.g. by advising a concept term that is more general or by aggregating preferences.

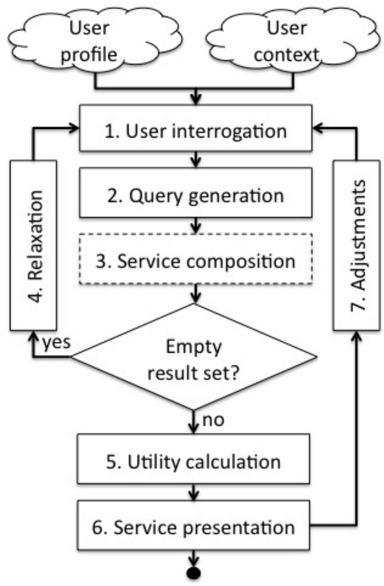

Figure 1: The process starting with the initial information (the user profile and context) ending in the final composition

Figure 1 illustrates our preference elicitation process. It starts with a questionnaire adapted to the user's expertise and context (step 1), which covers the following items: (i) the final goal of the Web service composition (a document, an action or an item), (ii) further specifications depending on the first answer, e.g. the type of the document, (iii) a demand for possible inputs by the user depending on the previous answers, (iv) properties to be regarded (e.g. price, time, quality) and (v) a menu to express the preferences over the chosen properties.

Items 1 to 4 can be realized with interdependent drop-down-menus or free text annotations depending on the user's expertise. Item 5 can be realized by letting the user construct a ranking of the properties or a numeral valuation of the properties. In step 1 not only an ordinal preference is evaluated but also the relation between the preferences. The further use of these results in step 5 is described below.

The questionnaire is used to gather new information, which (in combination with the previous information) is used for the query generation (step 2). The generated query is passed on to a service composition component (step 3) using only the hard constraints and the goal to find matching service compositions. The preferences are used for the ranking in step 6. In this paper, we consider a scenario with an implemented Web service composition component. This component is used for the first and for further iterations in the preference elicitation process. If the service composition returns an empty set, i.e. no services are found that match the query, the process repeats the information elicitation process with the aim of relaxing some requirements and preferences (step 4) like e.g. described in [Jan04]. To achieve this, the user may revise his former questionnaire or get proposals for relaxed preferences that he can choose from. Those proposals are computed in a way that they guarantee a non-empty result set if the user accepts them.

If at least one service is found, the utility of each Web service composition is calculated in step 5 based upon the information gathered in step 1. We propose two different approaches for the utility calculation: (i) using an actual utility function of the user or (ii) using fuzzy sets for calculating the overall acceptance (which is comparable to the utility of the service) like done in [ALS09]. For the former, a utility function u for the different attribute values v_{ij} of the according attribute a_i is recorded in the user profile and complemented in the preference elicitation process. Like in [TGL04] we consider a weighting for each attribute.The overall utility of one specific Web service composition c_j is calculated by summarizing all utility values of the Web service composition (cf. [ALS09]) adjusted with the weighting w_i of the respective attribute a_i: $U(c_j) = \sum_i w_i u(v_{ij})$.

If fuzzy sets are applied like e.g. in [AL05], the user has to specify categories for each attribute or choose according pre-defined categories for the attributes.[2] In [AL05] it was assumed, that the user could express his needs as fuzzy IF-THEN rules. When using fuzzy sets we cover the generation of these fuzzy IF-THEN rules and therefore the elicitation of the fuzzy sets, the (fuzzy) membership function of each attribute value to the according attribute category (both in step 1) and the calculation of the overall acceptance of a Web service composition (step 5).

The resulting values of step 5 may be used in the ranking of the results to be presented in step 6 where the composition with the highest utility value/acceptance is ranked first. The user history and context is used in this step, again, to adjust the result representation and explanations for the services. If the user is not satisfied with the result set presented to him, he may adjust or revise his previous statements (step 7). In this case a new questionnaire is presented depending on his desired changes or he may revise the previous questionnaire.

3 Related Work

Much research work (e.g. [ABK$^+$02, CP04]) dealt with automated eliciting preferences for product recommendations. Some approaches (e.g. [SPM06]) coped with applying preference customizing techniques for Web services and Web service discovery, but to our knowledge none actually used preference elicitation techniques, even though they bear many advantages to support Web service composition.

[2] Like [AL05] we consider only attribute values "which can be mapped to an interval scale". Nominal attributes, that are either fulfilled or not like e.g. a sepecific mode of shipping, need not be expressed fuzzy.

In [JK05] an application for a user preference elicitation system for e-services is described. In contrast to our approach the focus is not on direct service recommendations but on product recommendations in e-services . Nonetheless we adapt their idea to relax requirements if the result set is empty and their questionnaire adaption: The questions, answers and explanations are personalized in content, language and number according to the user's expertise. The defaults are set according to the user's expertise and previous answers. The final representation of the results and the explanation is adapted to his expertise.

In [PFT03] different aspects for decision support systems based on findings from behavior decision theory are stated. The authors claim that even though users are not capable of stating "other preferences beyond the very fundamental ones ... it is easier for them to critique examples, especially those showing violations" [PFT03]. We therefore conclude, that a revision of the preferences after a first result representation is crucial, so that the user can specify why some of the services presented to him are not fulfilling his needs.

The user model in [TGL04] covers user preferences that are derived by his tendencies, which are not solely indicated by his direct answers, but also by his decisions and a frequent provision of certain attributes or attribute values. It stores, for each item, the ratio of its acceptance to the number of times it was recommended to the user. In addition, with the user model the weighting for all attributes and the probability of the associated attribute values are stored. For initialization, the weighting is set according to the typical preferences in the use-case, e.g. in the case of a restaurant recommendation system the *price* has more importance and therefore a higher weighting than *parking*. The probabilities for the different attribute values start with a uniform distribution, while the accept/present-ratio begins with a large value, e.g. 90%. We anticipate to consider context information not only for the on-the-fly calculation but for the initial values as well.

In contrast to our approach, in [LHL97] the user preferences are handled as a set of constraints, which can also be weighted differently. The sum over all constraints C_i multiplied with their corresponding weights w_i results in the overall *error* of a proposed solution: $E((v_1, v_2, ..., v_n)) = \sum_{i=1}^{n} C_i(v_i) * w_i$ [LHL97].

4 Conclusion and Outlook

We presented a process model to elicit user information for Web service composition. Our approach uses information stored in a user profile as well as information derived from the user's context. Furthermore, preferences and hard constraints are surveyed via a questionnaire. Using a Web service composition component, all Web service compositions that satisfy the hard constraints build the result set. The collected information about the user's preferences allows implications on the user's utility function, which can be used to rank and present the services.

Future work will include concrete implementation details and evaluations. These will especially focus on the preference elicitation process itself, i.e. means for building a questionnaire to derive information about the preferences, the preference rankings, the according utility function and the implementation of the query generation.

Acknowledgements

This work is supported by the German Federal Ministry of Education and Research (BMBF) under the WisNetGrid-Project. We thank Sebastian Rudolph (AIFB, KIT) and René Jäkel (ZIH, TU Dresden) for their help.

References

[ABK+02] J.L. Ambite, G. Barish, C.A. Knoblock, M. Muslea, J. Oh, and S. Minton. Getting from here to there: interactive planning and agent execution for optimizing travel. In *18th national conference on Artificial intelligence*, pages 862–869. AAAI Press, 2002.

[AL05] S. Agarwal and S. Lamparter. SMART – A Semantic Matchmaking Portal for Electronic Markets. *Seventh IEEE International Conference on E-Commerce Technology (CEC'05)*, pages 405–408, 2005.

[ALS09] S. Agarwal, S. Lamparter, and R. Studer. Making Web services tradable: A policy-based approach for specifying preferences on Web service properties. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(1):11–20, 2009.

[CP04] L. Chen and P. Pu. Survey of preference elicitation methods. Technical report, Swiss Federal Institute of Technology in Lausanne, Technical Report No. IC/200467, 2004.

[Jan04] D. Jannach. Preference-based Treatment of Empty Result Sets in Product Finders and Knowledge-based Recommenders. In *Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence, KI 2004*, pages 145–159, 2004.

[JK05] D. Jannach and G. Kreutler. Personalized User Preference Elicitation for e-Service. In *IEEE International conference on e-Technology, e-Commerce and e-Service, Hong Kong, IEEE Computer Society*, pages 604–611, 2005.

[LHL97] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. *Courses and Lectures – International Centre for Mechanical Sciences*, pages 67–78, 1997.

[PFT03] P. Pu, B. Faltings, and M. Torrens. User-involved preference elicitation. In *IJCAI Workshop on Configuration*, pages 56–63. Citeseer, 2003.

[SPH04] E. Sirin, B. Parsia, and J. Hendler. Composition-driven Filtering and Selection of Semantic Web Services. In *In AAAI Spring Symposium on Semantic Web Services*, 2004.

[SPM06] S. Sohrabi, N. Prokoshyna, and S.A. McIlraith. Web service composition via generic procedures and customizing user preferences. *LNCS*, 4273:597–611, 2006.

[TGL04] C.A. Thompson, M.H. Göker, and P. Langley. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21(1):393–428, 2004.

[WS09] M. Winkler and A. Schill. Towards Dependency Management in Service Compositions. In *Proceedings of the International Conference on e-Business, Milan*, 2009.