Towards Optimizing the Non-Functional Service Matchmaking Time

Kyriakos Kritikos
ICS-FORTH
Heraklion, Crete, Greece
kritikos@ics.forth.gr

Dimitris Plexousakis
ICS-FORTH
Heraklion, Crete, Greece
dp@ics.forth.gr

ABSTRACT

The Internet is moving fast to a new era where million of services and things will be available. In this way, as there will be many functionally-equivalent services for a specific user task, the service non-functional aspect should be considered for filtering and choosing the appropriate services. The related approaches in service discovery mainly concentrate on exploiting constraint solving techniques for inferring if the user non-functional requirements are satisfied by the service nonfunctional capabilities. However, as the matchmaking time is proportional to the number of non-functional service descriptions, these approaches fail to fulfill the user request in a timely manner. To this end, two alternative techniques for improving the non-functional service matchmaking time have been developed. The first one is generic as it can handle non-functional service specifications containing n-ary constraints, while the second is only applicable to unary-constrained specifications. Both techniques were experimentally evaluated. The preliminary evaluation results show that the service matchmaking time is significantly improved without compromising matchmaking accuracy.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Systems—*Web-based services*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*; G.1.6 [**Numerical Analysis**]: Optimization—*Integer Programming*

Keywords

service, discovery, matchmaking, non-functional, QoS, optimization, performance, constraint programming

1. INTRODUCTION

The Web is now moving to a new era, called the Internet of Services (IoS), where millions of services and things will be available to users through a converged information, communication, and service infrastructure. Service-orientation enables such a change, as it is one of the main mechanisms exploited by application developers to build added-value applications through the discovery and composition of services.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

However, service-orientation has not yet delivered its promise as it relies on the role of the service broker, which is not successfully fulfilled by current realizations. Such realizations have failed due to the following factors. First, they are not scalable to handle thousands or millions of service requests and advertisements. Second, they exhibit low matchmaking performance and accuracy. Third, they have mainly concentrated on the functional aspect and have neglected the non-functional one. However, as there will be many functionally-equivalent services for a specific user task, the non-functional aspect should be considered for filtering and selecting the best among these services.

Concerning the functional aspect, many matchmaking approaches have been developed [3], mainly exploiting Semantic Web techniques and focusing solely on the service I/O. Such approaches exhibit good performance but their accuracy is not perfect. This is justified by the fact that the service functionality in terms of its pre- and post-conditions is not considered, partly because the service specifications (advertisements and requests) do not include the description of such an aspect. Due to the move to the IoS, specific approaches [4, 2] have concentrated on scalability issues and on further optimizing the service matchmaking time by appropriately organizing the service advertisement and request space. Such approaches move to the right direction but still face the challenge of imperfect accuracy.

While the research status concerning functional service discovery is satisfactory, the same cannot be stated for non-functional service discovery. The state-of-the-art approaches [1] in the latter sub-area exhibit perfect accuracy and mainly focus on optimizing the time required to matchmake a single non-functional request-to-advertisement pair by exploiting appropriate constraint solving techniques. However, they have not yet focused on optimizing the overall non-functional matchmaking time. As such, they can take significant time to match a set of hundreds or thousands of non-functional service advertisements against a non-functional service request, so they are not yet appropriate for the move to the IoS era. In addition to the above problem, there have not been approaches focusing on scalability issues.

Thus, there is a need for scalable techniques that are able to appropriately manage a vast amount of non-functional service offers and optimize their matchmaking time. If such techniques were coupled with those exploited for functional service discovery, then a better service broker realization would be offered, enabling users to discover those services matching their tasks both functionally and non-functionally in a timely manner and with the appropriate accuracy.

Two novel techniques were developed to close this research gap that significantly optimize the non-functional service matching time with respect to the techniques already proposed.

2. PROPOSED NON-FUNCTIONAL SERVICE MATCHING TECHNIQUES & EVALUATION

The first proposed technique, called "Subsumes matching" technique, relies on the "subsumes" type of matchmaking metrics and on the fact that if a non-functional service specification A subsumes another specification B then it will also subsume all the specifications subsumed by B. To this end, it organizes the non-functional service advertisement space in such a way that the number of non-functional request-to-advertisements pairs examined is less than that the state-of-the-art approach in [1]. In particular, it constructs a forest of "subsumes" trees, where each node corresponds to a non-functional service advertisement and a parent node in each tree subsumes all of its children nodes. In this way, when a service request is issued, it is compared against the nodes of each tree from the root until the leaves. However, if it is found that it subsumes a specific node, then there is no need to go further down to the node's children/descendants as the request will certainly match/subsume them.

This technique is obviously quicker than the state-of-the-art one, as each one uses the same matchmaking metric but the first technique performs less comparisons. However, the construction and update of a forest of "subsumes" trees is more costly than the construction and the update of a list of service advertisements (as it is the case for the state-of-the-art approach). To this end, this technique exhibits a higher insertion, deletion, and update time with respect to the state-of-the-art approach. This technique can be distributed by assigning the responsibility of matching a subset of the subsumes trees to different nodes.

The second proposed technique (i.e., the "Unary matching" one)appropriately organizes the non-functional service advertisement space based on the non-functional metrics / properties. In particular, it maintains for each non-functional metric an ordered set of limits, where each limit may correspond to one or more non-functional specifications containing a respective metric bound or equality constraint on the limit's value. To this end, when a non-functional service request is issued, each of its unary constraints are examined based on their containing metric. Depending on the metric bound and constraint type, a sub-part of the metric's ordered list of limits is examined so as to produce a list of the matching non-functional advertisements' URIs. For instance, if the request constraint is of the form: $X \leq a$, then the limits that are equal or less to a are examined and the URIs of the non-functional specifications that have constraints of the form $X \leq a_1$, or $X == a_1$, where $a_1 \leq a$, are collected. For each request constraint, its constructed URI list is concatenated with that of the previous constraint. If the URI list concatenation is empty, then the non-functional request does not have a matching advertisement. Otherwise, after the processing of the last request constraint, the final, concatenated URI list contains all the URIs of the advertisements that match the request.

The second technique is quicker than the other proposed technique as well as the prominent matchmaking approach not only in terms of matchmaking but also insertion, deletion, and update time. Moreover, due to the way it organizes the advertisement space, it can be easily distributed by assigning the responsibility of matching a sub-set of all non-functional metrics to different nodes. Its sole drawback is that it relies on exploiting only unary-constrained non-functional service specifications.

Both techniques were experimentally evaluated against one state-of-the-art technique. The preliminary results show that both techniques exhibit a better matchmaking time than that of the state-of-the-art one and that the second technique is significantly better and more scalable than the others.

3. CONCLUSION

This paper has proposed two novel techniques for optimizing the non-functional service matchmaking time. Moreover, it has reported some initial preliminary evaluation results which indicate that both techniques are significantly better than the state-of-the-art one and that the second proposed technique is more scalable.

Concerning future work, as there is a trade-off in optimizing both matchmaking and offer registration performance, research must concentrate on discovering the exact point where the performance in both aspects is satisfactory. In addition, future research should concentrate on further experimenting with the proposed techniques and appropriately integrating such techniques in functional service brokers. Moreover, additional techniques can be considered which intelligently organize also the service demand space. Finally, scalable and distributed, functional and non-functional service discovery mechanisms must be developed incorporating the two novel non-functional service matchmaking techniques. In this way, if the latter mechanisms are exploited by a service broker, then the vision of the Internet of Services will come closer to its realization.

4. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

5. REFERENCES

[1] K. Kritikos and D. Plexousakis. Mixed-Integer Programming for QoS-Based Web Service Matchmaking. *IEEE Trans. Serv. Comput.*, 2(2):122–139, 2009.

[2] S. B. Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, and Y. Berbers. EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. *J. Syst. Softw.*, 81(5):785–808, 2008.

[3] P. Plebani and B. Pernici. URBE: Web Service Retrieval Based on Similarity Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1629–1642, 2009.

[4] M. Stollberg, M. Hepp, and J. Hoffmann. A Caching Mechanism for Semantic Web Service Discovery. In *ICWS*, 2007.