2015 IEEE International Conference on Web Services

Discovering Web Services to Improve Requirements Decomposition

Hongbing Wang* Shuxiang Zhou* Qi Yu†
*School of Computer Science and Engineering and Key Laboratory of Computer Network and Integration, Southeast University
Nanjing, China, 211189
hbw@seu.edu.cn
†Rochester Institute of Technology, College of Computing and Information Sciences
Rochester, NY 14623, USA, qi.yu@rit.edu

***Abstract*—As a result of recent trends in enhancing Service-Oriented Requirement Engineering (SORE) activities, a number of requirement specification methods have been proposed for fitting the reuse infrastructure in a Service-Oriented Architecture (SOA). The availability of different Requirement Engineering methods offers developers a range of options to choose from. However, most of existing research effort uses traditional Requirement Engineering methods in service-based application developments. During requirements specification, a reusable infrastructure of available web services is not considered at all. The risk is that atomic requirements do not always fit reusable services. As a result, the service composition is time-consuming and needs costly adaption. This paper therefore proposes a novel method by introducing service discovery in the early Requirement Engineering stages so as to guide the requirement decomposition process. Although several researchers have already recommended to involve service discovery in SORE, they do not focus on how to guide requirement decomposition. Our approach is implemented on top of the widely used goal-oriented approach. To this end, we leverage a semantic service discovery method as a means to act as a guide and sentinel in requirement elaboration. We demonstrate the requirement decomposition process by implementing a case study from the Business Traveling domain.**

***Index Terms*—requirement decomposition; service discovery; goal-oriented approach; requirement engineering**

I. Introduction

Requirements of a software system describe the functionality offered by the system. Requirements decomposition is a core activity in which requirements are analyzed and decomposed to arrive at specific atomic requirements [13]. Specifically, in service based applications, complex requirements are decomposed for reusing relatively simple services. Using decomposition techniques that do not fit the characteristics of the system at hand will increase Requirement Engineering (RE) costs and make the system failure-prone. Hence, numerous decomposition techniques, as well as methods choosing appropriate techniques for a given system, have been developed and applied in practice for the development of service based applications [4] [7] [12]. Service based applications are usually in large scale, distributed, complex and involving loosely coupled entities owned by third parties known as web services. Hence, requirement decomposition in service based applications is crucial. Below we discuss some challenges of developing systems from reusing web services.

1) **The objective is reuse:** In service based applications, we decompose requirements so as to find reusable web services and combine them to establish user requirements.
2) **Dynamic environment:** As services may come and go, the feasible composition solution may change. This circumstance gives rise to a basic research challenge: how to decompose requirements dynamically.

To attack the above central challenges, some researchers have proposed the “Match early ” idea, which introduces service discovery in the early stages of RE [1] [2] [15].“Match early” is proved to perform well in industrial scenarios and has been employed in the development of service based applications [8]. However, a number of issues still remain to be addressed. This is because existing works mainly use match early to help complete the requirements or improve service discovery rather than guide/improve the requirements decomposition process. In this paper, we develop a novel requirements decomposition model which is also built upon the “Match early” mechanism but making key extension by leveraging KAOS [14] and semantic service discovery techniques.

Figure 1 shows the overall requirements decomposition process. Specifically, user requirements are modeled as root goals using KAOS in step 1. Condition 1 means at least one subgoal has no service left after filtering. Condition 2 means each subgoal has only one service left after filtering. Condition 3 means at least one subgoal has more than one services left after filtering. Then we elaborate each root goal in an iterative way and there are five steps in each cycle (step 2 to step 6). Finally we check the condition to find whether the decomposition succeeds or fails.

The remainder of this paper is organized as follows: Section 2 presents service-based requirements decomposition scenario that will be used throughout the paper to illustrate our work. Section 3 discusses related works. Section 4 presents the detailed requirements decomposition process. Section 5 presents the evaluation of our approach. Finally, section 6 concludes and outlines some future work.

978-1-4673-7272-5/15 $31.00 © 2015 IEEE
DOI 10.1109/ICWS.2015.52

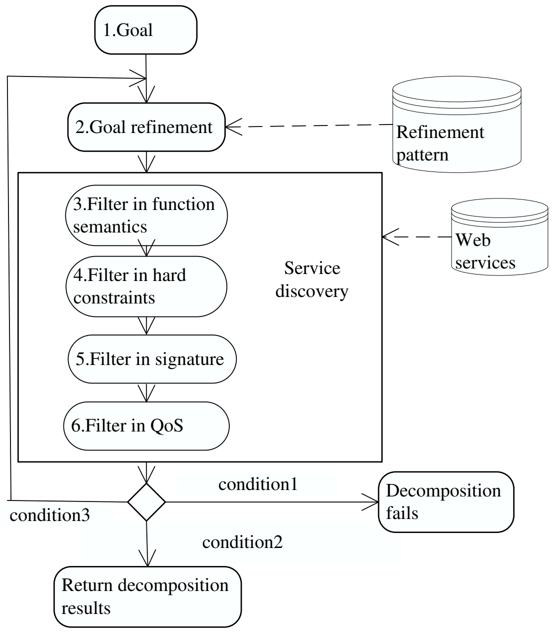

Fig. 1. Example Requirement decomposition

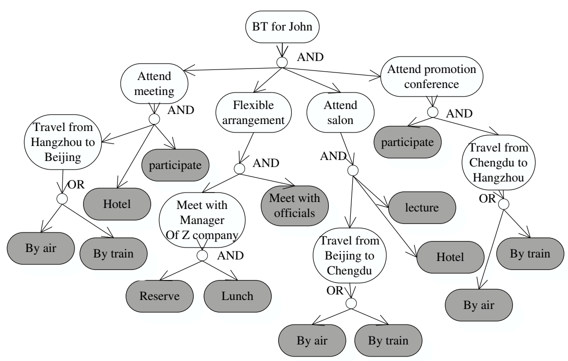

Fig. 2. Business Traveling plan for John

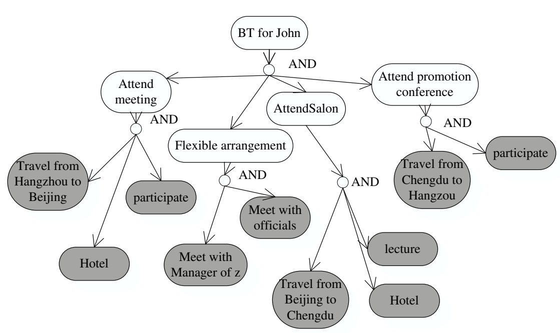

Fig. 3. Business Traveling plan for John specific to a given service set

II. An Illustrative Example

We describe an illustrating example, i.e., John's Business Traveling (BT), in what follows.

Mary is responsible for preparing a business traveling for her boss John. Here is the detailed request. John will attend an international meeting from May 8 to May 10 in Beijing. Besides, he will participate a salon in Chengdu on the morning of May 12 and a promotion conference on the morning of May, 14. During the free time between the meeting and the salon, he will meet with a manager of anther electronic company Z in Beijing and communicate with officers of the IT department. John wants to travel by air or by train and the hotel should be 5 star. We select this example for two reasons. First, business travel is a core activity of modern enterprises. This is a realistic and commonly used case. Second, a business travel plan requires lots of functionalities and exhibits various quality parameters. Without considering about reusable services, the decomposition will not end until all subgoals are atomic goals. In this case, the solution for John is shown in Figure 2. However, when we introduce service discovery early as illustrated in Figure 1, the result might be different. For instance, if there exist some complex services: a service "Travel" that can deal with traveling by air, by train and by bus; a service "LunchMeeting" that can send emails to partners and then make a reservation in a corresponding restaurant in advance. The decomposition process is shown in Figure 3. Comparing Figure 2 and Figure 3, it is unnecessary to further decompose goals "Travel from Hangzhou to Beijing", and "Meet with manager of company Z".

III. Related Work

Our proposed approach aims at integrating service discovery techniques to guide requirements decomposition. We outline a number of approaches that have tried to connect service discovery to requirements modeling. The authors of [5] provide means for transforming the process level service composition into AI planning problem. The work of [11] presents a service composition approach by bringing together the identification of application goals.However, it does not care about how to update and change the goal model at runtime.The work of [15] presents algorithms and evaluations of a novel service discovery approach. T.The work of [8] presents a tool-support approach to support requirements elicitation with the help of service discovery in the ERP domain. The work of [2] presents several advantages of match early approaches. However, above match early approaches focus on helping complete requirements or facilitating service discovery rather than improving the performance of requirements decompositions.

The work of [9] presents a service discovery approach which seems similar with our approach. After decomposition, service discovery happens, unfulfilled part of an original goal is extracted to be the new goal and discovery services iteratively. It is totally different from our work in this paper. The differences are shown as follows: 1) The objective, our work aims to improve the performance of requirement decomposition while their work focus on improving the performance of service discovery. 2) Our decomposition plan is independent of the order of matching while their work is dependent on the order of matching in discovery algorithms. 3) We consider about quality of services while they do not.

IV. The Model

The framework of our method has adopted an iterative process in which a goal is decomposed, partial candidates are then discovered, and finally these discovered services are used to verify whether the previous decomposition is feasible. We illustrate the process in what follows.

Goal to achieve *AttendMeeting*

InstanceOf *SatisfactionGoal*

Concerns *Meeting, schedule, hotel, travel*

InformalDefinition *John has to travel from Hangzhou to Beijing first and stay at a hotel before attending the meeting in time according to the schedule.*

Definition 1. ***Partial-candidate*** *is a web service which can provide all or part of the functions requested by a specific goal and satisfy the non-functional request of the goal.*

A. Goal refinement

In step 2 on Figure 1, goal G is elaborated into sub goals using patterns in KAOS. These subgoals are stored in a set called Sub-Goal. Sub-Goal(G)=$\{g_1, g_2, ..., g_m\}$. It means G could be directly decomposed into m subgoals ranging from g_1 to g_m.

B. Service discovery

In our approach, each goal has a related service set called CandidateSet containing its all partial-candidates. In Figure 1, after step 2, each subgoal's CandidateSet is initialized with CandidateSet of its parent. From step 3 to step 6, each subgoal's CandidateSet is narrowed by filtering unrelated services.

C. Filter in Functional semantics

A first portion of filtering services from a goal's CandidateSet is calculating Similarity on Description. Similarity on Description is based on general functional descriptions of a goal and a service. It aims to roughly filter those functionally irrelevant services specific for a goal. For instance, given a Business Traveling goal, by filtering in functional conceptual view, we can exclude services like online-education system, online-store and so on.

D. Filter in constraint

In a service, a constraint is presented as preconditions and effects.

E. Filter in signature

From a structural perspective, we use the typical paradigm of signature (I/O) annotated with ontology concepts. If we ignore the interactions during the execution and only care about the starting point and end point of a service, we get the signature (I/O) of it. For the match between a goal and a service in its CandidateSet on the view of signature, the desired relationship is 1:N. That is, if a service is a partial-candidate, it is kept; otherwise, the service should to be filtered out. First of all, we build a signature graph (SigG) using services left in its CandidateSet. Then, we find whether two services can be combined directly on signatures by building data type graphs. If yes, we add edges between them. After SigG is built, we find a valid path in SigG using graph algorithms.

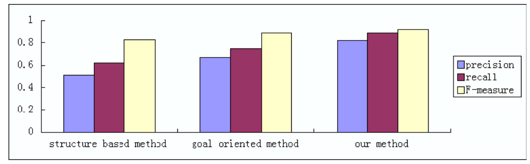

Fig. 4. Average precision, recall and F-measure of different methods

F. Filter in QoS

As we have already achieved a valid path, which is a possible decomposition plan of the goal. However, the plan might be incomplete because the decomposing is an iterative process. In this part, we check for global QoS attributes such as price, response time, and so on.

G. Check

If a goal has only one service left in CandidateSet, it is an atomic goal and the decomposition completes. Otherwise, if a goal has more than one services left, it needs further decomposition and goes to phrase 1 again. Anytime, if a goal has no service left in CandidateSet, the decomposition fails.

V. Experiments

We assess the effectiveness of our service based requirements decomposition approach in this section. The experiments are conducted using an IBM server with 8 CPUs of 2.13 GHz and a RAM of 16 GB.

A. Data Set

We use the public accessible QWS dataset [3] (http://www.uoguelph.ca/qmahmoud/qws/). It collects about 5000 web services, where each service is described as a WSDL document as well as QoS.

B. Comparative Result

To verify our approach, we involve three widely used web service composition cases and decompose their requirements on the above QWS data set using three different methods. The three methods are structure based decomposition, goal oriented decomposition as well as the proposed approach. Three cases are business traveling, weather forecast, and medical service. The result is shown in Figure 4. Our method acheives precision for 16% to 31%, recall for 13% to 27%. It means that considering available web services in requirement decomposition can improve precision. Precision in structure based method is the lowest since it lacks semantics.

id	use case
1	registration and logon
2	online test
3	score information searching
4	context management of text paper, question repository

TABLE I
ONLINE TEST

requirement	SF	I/O
R1	⟨ login, student or teacher, {{registrated id in database }, {three times fails, lock the account }} ⟩ ⟨ register, student or teacher, {{valid user },{null }}⟩	I: ID, password, role O: corresponding message
R2	⟨ test, student, {{test paper is available and test is done in requested period }, {forbid use of keyboard }}⟩	I: paper id O: test paper
R3	⟨ searching, student, {{only past four semesters are available }, {null }}⟩	I: semester O: grades
R4	⟨ setQuestionType, teacher, {{only five types within a test paper, basic three types unchangeable}, {opening type preferred }}⟩ ⟨ analysis, teacher, {{only past four semesters are available}, {null }}⟩	I: types O: report

TABLE II
TRANSFER USE CASE TO REQUIREMENT IN ONLINE TEST

C. Evaluation

We studied 15 real-world web based applications published by Ghazarian [6]. These systems cover diverse domain, such as online education system, online store, course evaluation system, and so on. We decompose these 15 requirements one by one using phrases mentioned in section 4. The results are that 11 cases succeed and the rest 4 cases fail. In 4 failed cases, project 5 and project 8 fail due to lack of services providing specific functions, project 9 and 10 fail due to violating QoS request.Then, we adopt the widely used standard definition of precision, recall and F-measure to verify the results. For each case, we calculate the average precision, recall and F-measure of all requirements in it. We take one of the 11 successful system programs as an example.The description of this Online Test program is shown in Table I.

We extract information from use cases and give the definitions of requirements in Table II. For each requirement from R1 to R4, we use the supposed approach to find a solution. We compute their precision, recall and only record the average results. We get a mean precision (and recall) of these 11 cases, which are beyond eighty percent.

VI. CONCLUSION

In this paper, we introduce a novel service based requirements decomposition approach. User requirements are modeled as goals, which serve as the starting point of our decomposition. Service discovery is introduced in requirements specification stage. The work presented in this paper is in its preliminary phrase and several possible directions are open for future work. First, our methodology needs to be more accurately evaluated by applying it to other realistic case studies. Second, we follow a "decompose and filter" mechanism, where the guidance of service discovery is still limited. Finally, we make the assumption that it is always possible to find a service composition exactly covering a goal so as to ensure a high successful rate . However, in reality, it may not be true since a goal sometimes can only be partially accomplished by existing services, hence requirement decomposition may fail. We plan to introduce a way to tolerate partial failure. That is, the decomposition of several subgoals fails while other subgoals success, which is similar to what has previously been done in [10].

ACKNOWLEDGE

This work is partially supported by NSFC Key Project (No.61232007) and Doctoral Fund of Ministry of Education of China (No.20120092110028).

REFERENCES

[1] Adam, S., Doerr, J.: The role of service abstraction and service variability and its impact on requirements engineering for service-oriented systems. In: 32nd Annual IEEE International Computer Software and Applications, 2008. COMPSAC'08. pp. 631–634. IEEE (2008)

[2] Adam, S., Uenalan, O., Riegel, N.: Is it beneficial to match reusable services earlier? In: Requirements Engineering: Foundation for Software Quality, pp. 136–150. Springer (2011)

[3] Al-Masri, E., Mahmoud, Q.H.: Investigating web services on the world wide web. In: Proceedings of the 17th international conference on World Wide Web. pp. 795–804. ACM (2008)

[4] Bai, L., Ye, D., Wei, J.: A goal decomposition approach for automatic mashup development. In: Enterprise Interoperability, pp. 20–33. Springer (2012)

[5] Bertoli, P., Pistore, M., Traverso, P.: Automated composition of web services via planning in asynchronous domains. Artificial Intelligence 174(3), 316–361 (2010)

[6] Ghazarian, A.: Characterization of functional software requirements space: The law of requirements taxonomic growth. In: 2012 20th IEEE International Requirements Engineering Conference (RE). pp. 241–250. IEEE (2012)

[7] Ghezzi, C., Menghi, C., Sharifloo, A.M., Spoletini, P.: On requirements verification for model refinements. In: 2013 21st IEEE International Requirements Engineering Conference (RE). pp. 62–71. IEEE (2013)

[8] Nöbauer, M., Seyff, N., Maiden, N., Zachos, K.: S3c: Using service discovery to support requirements elicitation in the erp domain. In: Advanced Information Systems Engineering. pp. 18–32. Springer (2011)

[9] Olsson, T., Chong, M.Y., Bjurling, B., Ohlman, B.: Goal refinement for automated service discovery. In: SERVICE COMPUTATION 2011, The Third International Conferences on Advanced Service Computing. pp. 46–51 (2011)

[10] Peng, Y.B., Zheng, Z.J., Gao, J., Jiang, X.Q., Ai, J.Q.: Method of two stages semantic service discovery. In: 2009 International Conference on Machine Learning and Cybernetics. vol. 3, pp. 1534–1538. IEEE (2009)

[11] Sawyer, P.: Satisfying requirements for pervasive service compositions. Models@ run. time (2012)

[12] Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D.: Htn planning for web service composition using shop2. Web Semantics: Science, Services and Agents on the World Wide Web 1(4), 377–396 (2004)

[13] Sommerville, I., Sawyer, P.: Requirements engineering: a good practice guide. John Wiley & Sons, Inc. (1997)

[14] Van Lamsweerde, A.: Requirements engineering: from system goals to uml models to software specifications (2009)

[15] Zachos, K., Maiden, N., Zhu, X., Jones, S.: Discovering web services to specify more complete system requirements. In: Advanced Information Systems Engineering. pp. 142–157. Springer (2007)