Towards Realizing Dynamic QoS-aware Web Service Composition

George Baryannis,1,2
1Department of Computer Science
University of Crete
GR 71409 Heraklion, Greece
{gmparg,dp}@csd.uoc.gr

Dimitris Plexousakis,1,2
2Institute of Computer Science
Foundation for Research & Technology - Hellas
GR 71110 Heraklion, Greece
{gmparg,dp}@ics.forth.gr

Abstract—In this paper, we identify two major issues related to Web service composition: the lack of formal specifications for services and service compositions and the inability of current service composition approaches to support dynamicity and QoS-awareness in an effective and scalable way. We analyze the underlying research challenges for each of these issues and propose a tentative research plan to address them.

Keywords—formal specification; service composition; QoS;

I. INTRODUCTION

The last decade has seen an exceptional wealth of research on topics related to Web services, especially by European researchers. The European Union 7th Framework Programme (FP7) has funded several service-related projects such as the Service Centric System Engineering Integrated Project (SeCSE), the Software Services and Systems Network of Excellence (S-Cube) and other projects collaborating under the auspices of Networked European Software and Services Initiative (NESSI).

Research on Web services covers a multitude of issues that are involved throughout the life-cycle of a Web service or a Service-Based Application (SBA). Among them, service description and service composition have attracted a great deal of attention from researchers. Automated Web service composition, in particular, has been a "silver bullet" in Web service research and many approaches that involve automation in the creation of the composition schema as well as during its execution have been proposed, with varying degrees of success. By conducting an extensive literature review, two major interrelated problems were identified: the lack of formal specifications for service compositions (or even for atomic Web services) and the inability of automated Web service composition approaches (especially the ones that employ AI planning techniques) to simultaneously satisfy requirements such as QoS-awareness, dynamicity and scalability in an effective way.

The existence of complete, formal, rich, semantic-aware specifications for Web services is of crucial importance for both service providers and service consumers. Service providers will be able to provide complete specifications of what they are offering, which would enable them to more effectively advertise their service products to potential clients. Service consumers, on the other hand, will be able to be informed of the exact way in which a service is expected to perform and the produced results, which will allow them to make informed choices and select the service that is the most suitable match for their requirements. By automatically processing specifications, service matchmaking can be achieved in a faster and less costly manner. Of course one needs to find the right balance between formality and tractability in processing specifications. Thus, formal specifications are one step towards the embracing of Service-Oriented Architectures by the industry.

While conducting the aforementioned literature review in automated Web service composition, we pinpointed a series of requirements that need to be met for an approach to be successful. These include dynamicity, semantic-awareness, QoS-awareness, non-determinism, partial observability, scalability, correctness, domain independence and adaptivity. Table I contains the results of a comparative analysis based on these requirements. The complete analysis is provided in [1]. As one can observe in the table, some of the requirements were satisfied by the majority of approaches, while others were rarely explored by the authors. To the best of our knowledge, no approach succeeds in satisfying the complete spectrum of the requirements we identified. Approaches that employ AI planning techniques, especially planning as model checking [2] seem to be more successful to that effect. At the same time, even this family of automated composition approaches hasn't succeeded in providing a means for dynamic QoS-aware service composition that is effective and scalable.

The rest of this document is organized as follows. Section II explores the research challenges associated with formal service specifications while Section III deals with those related to automated Web service composition. Section IV offers a preliminary plan for addressing the challenges that were identified and Section V concludes.

II. FORMAL SPECIFICATION OF WEB SERVICES AND SERVICE COMPOSITIONS

In this section, an analysis of the open issues concerning the description and formal specification of Web services and service compositions is carried out. These issues concern the ramification problem, the qualification problem and the problem of automatically deriving specifications for Web service compositions.
A. Addressing the ramification problem

In previous work [3], we identified the existence of the frame problem in Web service specifications that use the precondition-postcondition notation and proposed a solution inspired by work on procedure specifications. This solution was applied to the most prominent Semantic Web service description frameworks, such as OWL-S and WSMO. The frame problem is closely related to two other problems in logic-based knowledge representation, the ramification and qualification problems. In [4], the author defines the ramification problem as the problem of adequately representing and inferring information about the knock-on and indirect effects (also known as ramifications) that might accompany the direct effects of an action or event. The frame and ramification problems are somewhat contradicting: if one solves the frame problem by disallowing any effects except the ones explicitly stated, any solution to the ramification problem is precluded.

The ramification problem has not yet been examined in correlation with Web services and the effects it may have on their description and formal specification. In that context, one may deal with effects that are caused by effects of a service execution, i.e., executing a service affects the world in a certain way, which then leads to secondary effects that are derived from the primary ones. Finding a way to include such effects in a formal Web service specification while at the same time avoiding any conflicts with the aforementioned solution to the frame problem poses an interesting challenge.

Ramifications must be included in a service specification since the service consumer must be aware of all (direct and indirect) effects of the service execution. This is particularly important in service compositions, as the lack of knowledge of an indirect effect may lead to the assumption that a composition is valid and correct while that particular effect may contradict a precondition of another participating service, leading to an inconsistent composite service.

B. Addressing the qualification problem

While the ramification problem deals with the effects of an action, the qualification problem deals with the circumstances and conditions that must be met prior to the execution of an action. From its definition in [4] one can derive many correlations to the case of Web services. First of all, updating qualifications when new statements become part of our knowledge is something very common in the ever-changing world of Web services. This facet of the qualification problem is closely related to the adaptation qualities of a Web service framework. Hence, a solution to the qualification problem for Web service specifications would involve adapting such specifications whenever new knowledge is acquired, maybe due to new services becoming part of the repository at hand or due to the service provider modifying a specification.

Moreover, the qualification problem becomes very interesting in the case of composite service specifications. It is interesting to explore how the introduction of a new service in a composition schema (either as a replacement to an existing one that failed or by modifying the schema itself) affects the qualifications of services already participating in the composition. It is even more challenging to address such questions not at design-time, but at runtime, when the composite service is being executed and monitored.

C. Automatic derivation of composite specifications

The final challenge of this section serves as a bridge between service description and service composition. While service description frameworks attempt to describe service compositions using a variety of composition models ranging from orchestrations to choreographies to Finite State Machines, no framework (to the best of our knowledge) attempts to handle the problem of automatically producing specifications for a composite service, based on the specifications of the participating services. Likewise, no automated Web service composition approach attempts to derive a complete specification of the inputs, outputs, preconditions and effects (IOPEs) that should be provided to a service consumer.

Providing such complete specifications can be proven very useful since they can be used as complete formal descriptions of what the composite services are offering to consumers in terms of functionality and under what circumstances this functionality can be achieved. This will promote and facilitate the reusability of composite services. Such specifications would also be of great assistance when one attempts to deduce whether a set

---

1 A ✓ denotes that the particular requirement has been addressed by multiple approaches in the category. A ~ denotes that the particular requirement has been addressed by some approaches in the category.
of services can actually be composed in a meaningful way. If inconsistencies are detected during the process of automatically creating the composite specifications, then the set of services is not composable and a replacement should be found for the service or services that cause the inconsistencies.

Automatically deriving the IOPEs of a composite service is directly linked to the way services are orchestrated. A trivial solution would be to expose the full set of preconditions and postconditions of all participating services. This, however, wouldn’t scale well to large compositions consisting of many complex services. Thus, a middle-ground solution should be explored: composite specifications have to reveal some part of the specifications of some of the internal services. The challenge is to decide what should be included and what should be left out.

III. AUTOMATED WEB SERVICE COMPOSITION

In this section, we will briefly summarize the research challenges that were derived from the state of the art survey and comparison we conducted, the results of which are shown in Table I. Notice that we exclude any discussion about the requirement of adaptivity. We consider service adaptation to be a major research area with many interesting challenges, hence it needs to be dealt with separately from our research agenda, which concerns service composition.

A. QoS-Awareness in AI Planning techniques

When examining Table I with regard to QoS-Awareness, one can deduce that while it has been explored (adequately or not) in several categories of approaches, it has been largely ignored in the case of AI planning. The only AI planning effort in our state of the art survey that deals with QoS is the work of Zeng et al. [5]. This work, however, fails to meet very important requirements such as support for nondeterminism and partial observability as well as scalability and domain independence.

Introducing QoS-awareness in AI planning techniques involves making at least three fundamental decisions. The first decision is to choose a planning technique which will then be adapted to be QoS-aware. The comparison table can again be useful to making that decision as we can choose a technique that has shown promising results in meeting the rest of the requirements that we have set for automated Web service composition. Planning as model checking is one such technique, as it is the only one that addresses the issues of nondeterminism and partial observability.

A second decision involves the QoS model that will be employed. Rather than creating one from scratch, a preexisting one could be used, as long as it satisfies some important requirements such as being expressive, formal and fine-grained, as well as being based on Semantic Web concepts. Another important aspect is to decide on which phases of the composition process the QoS characteristics will be applied and the effects of that application. For example, introducing non-functional characteristics in AI planning techniques may assist in speeding up the plan generation by limiting the search space when excluding services that don’t meet preset QoS thresholds. However, it may also do the exact opposite if we include QoS criteria in the plan goal, since it will make the goal more complex and harder to satisfy. It is apparent that introducing QoS-awareness is a rather challenging task.

B. Dynamicity in AI Planning techniques

As is the case with QoS-awareness, dynamicity is another requirement that hasn’t been adequately explored in AI planning techniques. The vast majority of them produce a static composition schema (the generated plan) without exploring the case of run-time composition where composition schemas are abstract and are only made concrete at run-time. The most prominent exception is the work of Klusch et al. [6] which achieve some level of dynamicity through the use of replanning. Peer uses it to recursively generate plans that are incrementally closer to the required goal. This introduces dynamicity to the composition process but only at design time.

On the other hand, Klusch et al. use replanning whenever an agent detects that a service’s preconditions have been violated during the execution of a plan. The replanning module is informed of the position of the error at the plan and tries to fix the problem by searching for an alternative path in the connectivity graph from that position onwards. To reduce the search space, unnecessary services can be blocked to avoid a complete preprocessing phase.

Klusch et al.’s effort is a step in the right direction, however, as is usually the case, it lacks many other important features, such as support for nondeterminism and partial observability, QoS-awareness, scalability and any proof of correctness. It should be challenging to explore how dynamicity at run-time via replanning can be applied to planning techniques that support most of these features, such as planning as model checking. Thus, the advantages of both approaches could be combined to result in a composition framework that achieves most of the composition requirements we have set.

C. Scalability

Another requirement that needs to be satisfied by an effective automated composition framework is that of scalability. Some of the approaches that we examined explicitly mention scalability and provide details of the authors’ efforts to address it, supported by experimental evaluation, but the majority does not provide such comprehensive examination of the problem while others simply ignore the issue.

While examining how a composition approach scales, one should explore the main causes that have a negative effect on the efficiency of the approach and try to remove them. However, this should be done only after evaluating the significance of limiting the approach in that way. Achieving scalability but at the same time crippling the overall applicability of the approach is obviously unreasonable since the advantage offered by the former is
canceled by the effects of the latter. In any case, a complete composition approach should include a detailed account of the way efficiency is affected when the complexity of the composition problem is increased.

IV. RESEARCH PLAN

Based on the research challenges defined in the previous section, we have identified two major research goals that make up our research plan. The first goal is to define and formalize a novel specification language for Web services and service compositions. This language will take into account the frame, ramification and qualification problems and offer robust solutions based on its foundations, similarly to the way Kakas et al. [7] provided an action language that solves the aforementioned problems by the way it is defined. Moreover, this language should provide support for the derivation of composite specifications based on a composition schema.

The initial milestone we have set regarding this first goal is to explore the challenge of automatically deriving composite specifications, given the composition schema and the specifications of the services taking part in the composition. We are in the process of identifying the possible values for preconditions and postconditions in sequential and parallel compositions and calculating the weakest possible sets of them by drawing inspiration from the field of programming language specifications and the Craig-Lyndon interpolation theorem [8].

A second milestone involves identifying the cases in which the frame, ramification and qualification problems appear, to examine the effects it may have and to assess the severity of them. Then, using the insight gained by these previous steps, we will be able to define the aforementioned service specification language.

After the first goal is achieved we can move forward to the second research goal, which is the design and implementation of a dynamic, QoS-aware automated service composition framework. This framework will rely on the results of the first goal in the sense that it will expect services to be specified using the aforementioned specification language and will use the same language to describe the composite services that will be produced. A tentative road map for this goal is to begin with the introduction of QoS characteristics in the specification language by exploring the work of Kritikos and Plexousakis [9], which extends OWL for QoS-based Web Service Description and Discovery. A further milestone will be to select a successfull planning technique such as planning as model checking and attempt to introduce dynamicity by applying run-time replanning methods.

V. CONCLUSIONS

In this paper, we identified two major problems in the field of service composition: the inability of current service description and composition frameworks to provide complete formal service specifications and the lack of an effective Web service composition approach that combines most desired requirements, such as QoS-awareness, dynamicity and scalability. The underlying research challenges were also analyzed and a tentative research plan to address them was outlined.

The importance of formal specifications for Web services and service compositions is crucial for the realization of the goals of Service-Oriented Architecture and should prove beneficial to both service providers, who will be able to better describe what they are offering and to service consumers, who will have a complete knowledge of what a service does under certain circumstances. On the other hand, realizing scalability, dynamicity and QoS-awareness in automated Web service composition may prove to be the necessary incentive for the industry to consider adopting such techniques for the composition of Web services. Of course, the eventual adoption of such an approach by the industry also depends on the availability of semantically-enabled Web services and the existence of effective discovery techniques for such services.

ACKNOWLEDGEMENT

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).

REFERENCES


