



Service Composition Scenarios in the Internet of Things Paradigm

Pawel Stelmach

► To cite this version:

Pawel Stelmach. Service Composition Scenarios in the Internet of Things Paradigm. 4th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Apr 2013, Costa de Caparica, Portugal. pp.53-60, 10.1007/978-3-642-37291-9_6. hal-01348735

HAL Id: hal-01348735

<https://hal.archives-ouvertes.fr/hal-01348735>

Submitted on 25 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Service Composition Scenarios in the Internet of Things Paradigm

Paweł Stelmach¹

¹ Wrocław University of Technology, Institute of Computer Science, Wybrzeże
Wyspiańskiego 27, 50-370 Wrocław, Poland
Pawel.Stelmach@pwr.wroc.pl

Abstract. In this work the topic of service composition scenarios is introduced and discussed with the use of leading example of service composition in the transport domain. The example is used to visualize the connection to the Internet of Things paradigm. The service composition itself is defined with a holistic approach, describing all steps of the proposed composition process like: requirements definition, requirements decomposition or aggregation using domain ontology, composite service structure construction, service discovery, structure and service plan optimization. Lastly, implementation of service composition with service composition scenarios is discussed.

Keywords: service composition, Service Oriented Architecture, service composition scenarios.

1 Introduction

The demand for quick delivery of new functionalities to applications is increasingly more significant in today's world. Sometimes it can determine the future of a big company that cannot quickly adapt to new market requirements. To give an example, one can think of banks that needed to add Internet capabilities for their clients and their monolithic, enormous applications put them in a disadvantage, compared to new Internet banking companies.

Web services have received much interest due to their potential in facilitating seamless business-to-business or enterprise application integration, allowing for building smaller component-based applications that could be easily replaced or extended with new functionality.

When applications are distributed a new problem arises – how to connect their capabilities. One could hardwire services but this, assuming existence of hundreds of services in a company, would inevitably lead to problems with clarity of so designed systems. Functionalities obtained via service compositions allow for loose coupling of services, connecting them via their interfaces and allowing for easy interchangeability. Much work on the use of web services has been done in the context of Service Oriented Architecture paradigm [1].

Still, the trend of outsourcing more and more tasks to outside companies and personalization of systems, created a need for more automation and thus, focusing interest on automated service composition. In the last decade many researchers have

presented their approaches to service composition. Basic approaches presented the use of AI Planning techniques ([2]) or, on top of it, introduce other ideas like two-staged composition with abstract processes ([4]) and more recently [5]. Through the last years various other approaches were introduced, among them workflow-based ([6]) and template-based approaches ([7], [8]). More recent take on service composition is visible in emergence of semantic match making in service composition ([9]) and fulfillment the Quality of Service (QoS) requirements ([10], [11], [12], [13]).

This paper focuses on another perspective. Rather than analyzing different methods for building abstract composite services, it is investigated what really are the requirements for the service composition and how a composite service can be obtained. To do this we must begin with the analysis of the leading example of service composition, namely, composition in the transport domain and present approaches of some other authors to this subject.

The remainder of this paper is structured as follows. In Section 2 relation to the Internet of Things is discussed. Section 3 defines service composition problem definition and motivation for research. Section 4 presents research contribution and innovation. Section 5 discusses the applicability of proposed innovations. Finally, in section 6 concluding remarks and future research plans are presented.

2 Relationship to Internet of Things

Web Services are crucial to the Internet of Things paradigm. They allow for precise definition of capabilities of interfaced objects and interaction with them. Appropriate web description of real world objects will allow Internet users to define their requirements of those objects and their capabilities. Those requirements could be increasingly complex, leading to defining requirements of composite interactions between specific objects or, generalizing, requesting strictly capabilities and not specific objects. Such process of designing requirements and finding optimal selection of objects available via web services could be defined as service composition problem.

In transport domain applicability of web services and service composition to the Internet of Things paradigm can be twofold. Typically, services could be divided into travel search and reservation services. However, in the context of the Internet of Things paradigm Web Services and service composition can look quite different than a typical operations-based service composition. If, in the travel domain, we define a task of travel composition then one could use services that interface real planes or flights to be taken and a sequence of flights from consecutive cities could be looked at as a composite service that consists of a series of Web Services. In the context of Internet of Things each flight or plane could be responsible for its own information management, registration etc., making it more natural to have separate services for each “thing” in the world. However, regardless of the choice of approach, both models recognize the existence of real world objects and define Web Services that mediate their use – whether it is a single service for one object or managing a set of objects of one owner.

3 Problem Definition

3.1 Service Composition

The service composition problem can be presented as transformation of user requirements into a composite service execution plan that fulfills them (Fig.1). Typically user will present both his functional and non-functional requirements, sometimes referred to as Service Level Agreement (SLA). To find a service that fulfills those requirements a three-stage approach is proposed:

- **structure:** SLA with all required functionalities is defined and given a structure of a graph, where each node contains a functional requirement and edges that connect graph nodes define the required control flow,
- **scenario:** the set of edges of the graph is extended so that the graph is consistent (complete information about order of execution of all functionalities is known) and service repository is searched in order to find services capable of providing requested functionalities; the main goal of this stage is to find services that fulfill each of the requirements but may differ in non-functional parameters (this part is usually performed using semantic filters)
- **optimal execution plan:** for each vertex representing a required functionality only one service is selected (from candidates gathered in the previous step), in such a manner that all the services in the structure fulfill the non-functional requirements (this is determined according to the aggregated QoS parameter value of the whole composite service)

More details on the three-stage service composition and algorithms for this approach can be found in [14].

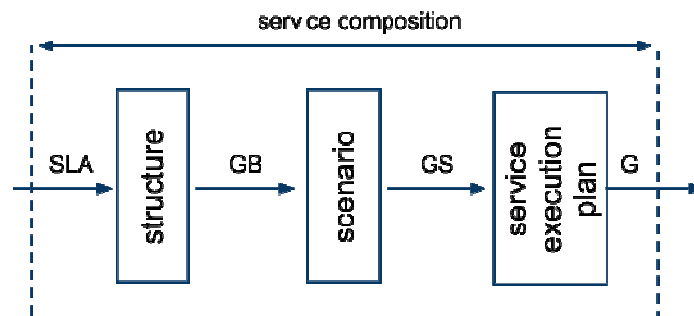


Fig.1. Stages of service composition approach.

3.2 Motivation

What is a service? In aerial transport domain, is it a single flight from A to B or a flight-searching component that is connected to a database of “A to B”-like flights. A programmer would more likely choose the latter, however, in the Internet of Things paradigm it is not so obvious. The distinction is deeply dependent on when and how would we want to use such a service and, getting closer to the topic of service composition, what would a composite service look like and how or when would we want to compose it. Those questions are very often omitted in literature and authors propose a real life motivating example but quickly switch to abstract ideas mixing all types of composition approaches and neglecting the notion of describing in which situations composition algorithms should be used.

For example in [15] (see Fig. 2) a composite service is comprised of a TravelPlanner service and services like AirlineBooking, HotelReservation and CarRental. However, this shows how often two different roles are mixed together: of a designer and a final user. The designer does not need the automated composition to create such a service and the user does not want to know that a TravelPlanner service exists and will not demand it. Thus, no one needs such composition. This happens because TravelPlanner service and HotelReservation service are not of the same kind and it should not be the goal of service composition to achieve such a service in one end-to-end process at a single time.

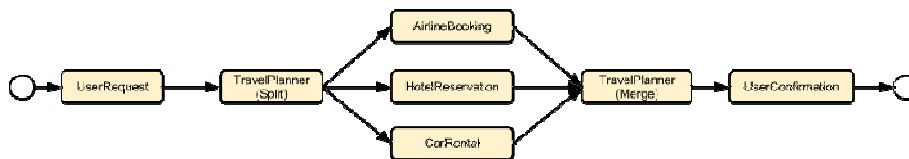


Fig 2. Travel Planner composite service.

4 Research Contribution and Innovation

4.1 Client and Provider Perspective Distinction

From the client perspective the actual focus is put on the *result* of the requested service and not the service itself. When we imagine our clients' actions, they do not originate in a Google-like web service search engine. In fact it is more probable that the client already is on a travel-planning site and so the service composition request is not at all random and in should not be treated as generic with a generic composition approach. In this context we should rather empower the provider with the right tools to build appropriate, simple and domain-specific web forms, rather than to use some kind of generic tools e.g., graph-based composition user interface to generate requests similar to one presented in Fig.2.

When describing the providers' perspective, it is necessary to understand that the composite service provider does not have to be the provider of web services used in the composite service. In fact, he could be seen as an intermediary between a client and the actual web service provider. His goal is to create such a service that the client would want to use and to that purpose he would typically use manual composition approach. He would not want to automatically compose a service as in Fig. 2 and definitely would not want to burden the user with such a task or risk that the composition algorithm will omit crucial parts of the composite service like TravelPlanner service. However, he does not want to create and maintain dozens of possible composite services that differ in the selection of airline, bus, train, car or hotel reservation depending on what are the users' current needs. What he would like is to *manually* define some kind of template that would want to allow a controlled composition to be dynamically performed for the user, while determining the limits and behavior of such a controlled composition mechanism.

4.2 Service Composition Scenario Definition

For this purpose we present a scenario driven dynamic and data driven service composition approach. Scenarios, that define the scope of the composition, are defined manually by the composite service provider along with the basic template describing what parts of the composite service should remain constant.

Fig. 3 presents such a scenario where the user actions and travel planning services are statically bound at design time and other parts of the composite service are represented as dynamic composition requests that will result in the composition and execution of a composite service so that the service composition is dependent on the data received from the previous step.

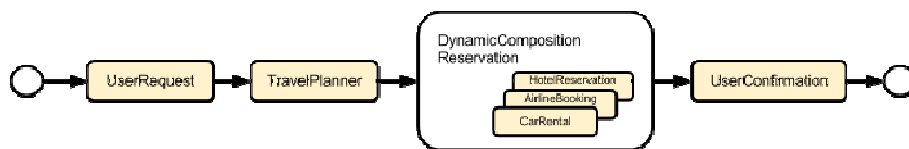


Fig.3.An example with a composition scenario.

The stages of dynamic composition can be further controlled in the service composition scenario with:

- limiting the set of candidate services,
- choice of a single-step approach, selecting Web Services directly, as in AI Planning techniques or a choice of a two-step approach, first selecting abstract services and then performing dynamic service discovery and binding,
- selecting a goal-based technique over a keyword-based technique,

- if possible, preferring parallel structures over series or series when some services need data from the previous services and should be composed in such a way.

With such approach the composite service provider can deliver a *single* composite service that will be partially recomposed at runtime, assuming that he can deliver appropriate data with the UserRequest or TravelPlanner service.

4.3 Advanced Composition Scenarios

In previous sections, in order to avoid confusion, a simple example taken from literature was used. However, even that example is in fact not what the composite service provider would want. The TravelPlanner service would have to have access the data from all providers i.e., airline, hotel and car provider or more if the user would want to use train instead of a plane. Also, the data would be accessed via Web Services and this would stand in contradiction with the SOA paradigm that services should not execute other services directly. In fact, the travel planner should be a composition of search services, dynamically composed based on users requirements (see Fig. 4). Only then a merging of results may happen and a suggested travel plan that is valid in each option should be delivered to user interface.

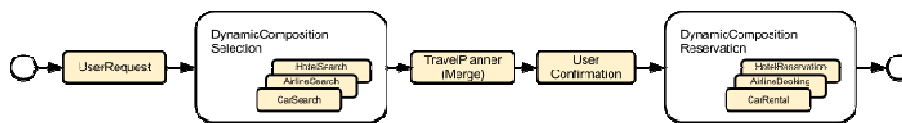


Fig.4.Example of an advanced composition scenario.

The example in Fig. 4 also shows that the difference between selection and reservation could be easily neglected when designing a domain example, however, it presents itself with important implications such as wrong placement of the user confirmation service (in Fig. 2 it is *after* the reservation services). And if the human operator can get it wrong, how the automated composition could know it.

5 Discussion of results

In the motivation in section 3 it has been briefly mentioned that it is possible to compose services using them as immediate interfaces for singular real world objects (namely one service corresponds to one object or operation like flight from A to B). However, most examples presented a typical approach with generic services. This does not mean, of course, that the general-purpose services don't implement the Internet of Things paradigm. They still do but in such a scenario they are responsible for sometimes hundreds of such real world objects.

Still, it would be valid to present a service composition of such “atomic” services. One could use composition scenarios presented in previous sections and in a dynamic composition component use AI Planning techniques for composition of a series of “A to B” travel services. However, to bring back the focus of this paper, we should consider why such a composition should be delivered. As an effect of such considerations one has to remember that for the user the reason for using service composition is in fact its results and, indirectly, its execution. Here, the execution of the travel service itself should be delayed in time and prior to that the reservation should be performed. Such composition could be useful with composing or recomposing long running processes, when the user is in fact “inside” the process as it executes but this topic is beyond the scope of this paper.

6 Conclusions and further work

In this paper a topic of service composition in the Internet of Things paradigm was discussed. A motivating example for the need of different approach to service composition was introduced and, in that context, service composition scenarios based approach was presented and discussed.

Further work will focus on detailed definition and systematization of various composition mechanisms to structure, scenario and optimal plan stages and their adaptation to the composition scenarios format.

Research presented in this paper will be included in the PlaTel platform for composition and management of telecommunication services [16] as part of the composite service execution engine dynamic behavior component. This research is a part of ongoing research on service composition and its application to various domains like telecommunication, sports or transport.

Acknowledgments. The research presented in this paper has been co-financed by the European Union as part of the European Social Fund.

References

1. SOA Reference Model Technical Committee. A Reference Model for Service Oriented Architecture, OASIS(2006)
2. Ponnekanti, S. R., Fox, A.: SWORD: A Developer Toolkit for Web Service Composition. In: Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA (2002)
3. Aggarwal, R., Verma, K., Miller, J., Milnor, W.: Constraint Driven Web Service Composition in METEOR-S. In: Proceedings of the 2004 IEEE International Conference on Services Computing, 23 – 30 (2004)
4. Xin, L., Xinhuai, T., Zhaoteng, S., Xiaozhou, Y., Delai, C.: AFlow: An Automated Web Services composition system based on the AI planning and workflow. In: Proceedings 2010 International Conference on Progress in Informatics and Computing, 1067- 1071 (2010)

5. Wu, Z., Ranabahu, A., Gomadam, K., Sheth, A. P., Miller, J. A.: Automatic Composition of Semantic Web Services using Process and Data Mediation. In: Proceedings of the 9th Intl. Conf. on Enterprise Information Systems (2007)
6. Pathak, J., Lutz, S. B. R., Honavar, V.: MoSCoE: A Framework for Modeling Web Service Composition and Execution. In: 22nd International Conference on Data Engineering Workshops, IEEE Computer Society (2006)
7. Sirin, E., Parsia, B., Hendler, J.: Template-based Composition of Semantic Web Services, In: AAAI Fall Symposium on Agents and the Semantic Web, Virginia, USA (2005)
8. Jie, G., Bo, C., Junliang, C., Lei, Z.: A Template-Based Orchestration Framework for Hybrid Services. In: 2008 Fourth Advanced International Conference on Telecommunications, Athens, Greece. p. 315-320 (2008)
9. Klusch, M., Fries, B., Sycara, K.: OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services. *Web Semantics: Science, Services and Agents on the World Wide Web* 7, 121–133 (2009)
10. Anderson, S., Grau, A., Hughes, C.: Specification and satisfaction of SLAs in service oriented architectures. In: 5th Annual DIRC Research Conference, pages 141–150 (2005)
11. Ko, J. M., Kim, C. O., Kwon, I.-H.: Quality-of-service oriented web service composition algorithm and planning architecture. In: *Journal of Systems and Software*, vol. 81, no.11, pp. 2079-2090 (2008)
12. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for Web services selection with end-to-end QoS constraints. In: *ACM Transactions on the Web*, vol.1 no.1, p.6-es, May (2007)
13. Blanco, E., Cardinale, Y., Vidal, M.-E., Graterol, J.: Techniques to Produce Optimal Web Service Compositions. In: *IEEE Congress on Services*, 553-558 (2008)
14. Stelmach, P., Grzech, A., Juszczyszyn, K.: A Model for Automated Service Composition System in SOA Environment. In: Camarinha-Matos, L.M. (ed.) *DoCEIS 2011. IFIP AICT*, vol. 349, pp. 75–82. Springer, Heidelberg (2011)
15. Jeong, B., Cho, H., Kulvatunyou, B., Jones, A.: A multi-criteria web services composition problem. 2007. IRI 2007. *IEEE International Conference on Information Reuse and Integration*. IEEE (2007)
16. Świątek, P., Stelmach, P., Juszczyszyn, K., Prusiewicz, A.: Service Composition in Knowledge-based SOA Systems. In: *New Generation Computing* 30.2: 165-188 (2012)