

# A UML-based Rich Service Description Language for Automatic Service Discovery of Heterogeneous Service Partners

Zille Huma<sup>1</sup>, Christian Gerth<sup>1,2</sup>, Gregor Engels<sup>1</sup>, and Oliver Juwig<sup>3</sup>

<sup>1</sup> Department of Computer Science, University of Paderborn, Germany\*\*

{zille.huma,gerth,engels}@upb.de

<sup>2</sup> s-lab - Software Quality Lab

{cgerth}@s-lab.upb.de

<sup>3</sup> HRS-Hotel Reservation Service, Germany

{Oliver.Juwig}@hrs.de

**Abstract.** Service-oriented computing (SOC) emerges as a promising trend solving many issues in distributed software development. Following the essence of SOC, service descriptions are defined by the service partners based on current standards, e.g., WSDL [15]. However, these standards are mostly structural and do not provide any behavioral description, which may lead to inaccurate service discovery results. There is a requirement for a *rich service description language* for service partners that encompasses the structural as well as behavioral information in the service description. Furthermore, service discovery based on an automatic matching of these comprehensive service descriptions is a complex task, which is further complicated through the heterogeneity of the service partners' domains in terms of different underlying ontologies. In this paper, we propose a rich service description language based on UML, which allows the specification of structural and behavioral features of a service. In addition, we also briefly discuss how some existing matching approaches can be extended to define an automatic matching mechanism for rich service descriptions resolving the underlying heterogeneity.

## 1 Introduction

Service-oriented computing (SOC) realizes the idea of reusability through independently developing, automatically discovering and consuming distributed software components (services) on the basis of their interface descriptions.

The automatic discovery of services at design-time as well as at run-time faces certain challenges. The first challenge is that of insufficient information in service descriptions. Standards for *structural* description, such as the Web Service Description Language (WSDL) [15], are unable to comprehensively describe the service *offer* or *request* of service partners. Semantic web service (SWS) approaches [8, 12, 2] come up with notations for rich description of the service

---

\*\* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901)

offer/request. However, these approaches face certain limitations. For instance, they are not comprehensive enough to cover multiple aspects of service description, such as operation semantics or service protocols, etc. Similarly, [12, 2] are not yet widely accepted in practice because of their diversion from the existing standards and the extra effort required to learn and use their complex notations.

The second challenge is the heterogeneity of service descriptions. The distributed paradigm allows service partners to function in their respective domains and describe their service *offer* and *request* conforming to their own enterprise model (ontology). As a consequence, service descriptions may be semantically similar but are specified differently conforming to their individual ontologies. For automated service discovery such similarities must be identified. Moreover, due to the underlying heterogeneity, the correspondences between the provided and the required operations in the service descriptions may not be limited to 1:1 but there may be more complex types of correspondences, e.g. 1:n, n:1, or n:m, which additionally complicates the matching of service descriptions.

As a solution that addresses these challenges, we propose a UML-based rich service description language in this paper. We give an insight into the features of the proposed language and briefly discuss how the existing matching approaches can be extended to formulate a matching approach that includes heterogeneity resolution features as well. The remainder of this paper is structured as follows: In Section 2, we introduce a real-world scenario, discuss its limitations and outline detailed requirements for a potential solution. Section 3 describes the proposed rich service description language in detail. Section 4 discusses the existing matching approaches with their shortcomings and outlines a potential solution. In Section 5, we discuss related work. Finally, in Section 6, we discuss future directions of our work.

## 2 Example Scenario and Detailed Requirements

Figure 1 shows a typical scenario of SOC, where service requestors define their *service request* (SR) and service providers describe their *service offer* (SO) based on their independent local ontologies. These SRs/SOs need to be matched for the purpose of service discovery.

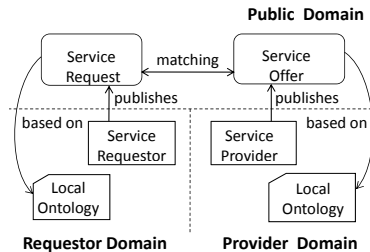


Fig. 1. Typical Scenario of SOC

As a simplified example scenario, we consider a case study from the e-tourism domain, where tourists can book flights and accommodations online. Figure 2 shows a typical reservation scenario provided by the worldwide accommodation reservation company, Hotel Reservation Service (HRS)<sup>4</sup>. Users connect with HRS through a variety of interfaces like web browser or smart phone. On the other end, HRS as a *service requestor*

<sup>4</sup> <http://www.hrs.com>

connects with services of their partner hotels that act as *service providers*, to carry out the booking of accommodations.

Whenever HRS wants to extend the business by connecting to a new hotel or hotel chain, HRS’s web application has to connect to the provided services of new partner hotels through manual matching of SR of HRS and SOs of the hotel/hotel chains.

Currently, the SR of HRS is based on structural features, i.e., an operation signature description conforming to the tourism ontology by the Open Travel Alliance (OTA)<sup>5</sup>. Figure 3 (a) shows an excerpt of this local ontology. An example SR may contain the specification of the following operation signatures: *checkAvailability()*, *getDetails()*, *makeReservation()*, whose input and output parameters are typed over the concepts contained in this local ontology.

To fulfill the requirements of the SR, we assume that there exist two potential service providers *HotelX* and *HotelY*, whose service offers (SOs) are based on the HarmoNET<sup>6</sup> tourism ontology, which covers major sub-domains in e-tourism similar to the ontology by the OTA. Figure 3 (b) shows the HarmoNET-based local ontology for *HotelX* and *HotelY* extended with booking-related concepts. The structural SO of *HotelX* comprises the following operation signatures: *getAvailableRoom()*, *makeABooking()*, and *generateReceipt()*. Similarly, structural SO of *HotelY* has the following operation signatures: *searchRoom()*, *getRoomDetails()*, *bookRoom()*, and *getReceipt()*.

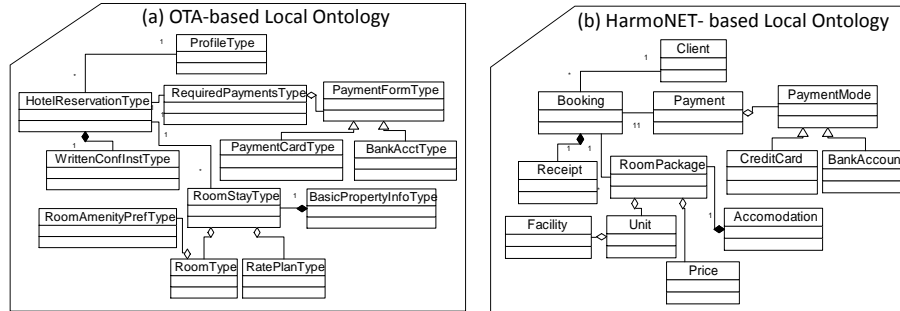
These SOs of *HotelX* and *HotelY* specifying the structural features are manually matched to the structural SR of HRS for service discovery. The service matching is further supported through personal communication between the service partners based on natural language and UML-based diagrams. Such a manual matching based on structural service descriptions makes the service discovery process time-consuming and expensive in terms of time and resources. Additionally, it is also error-prone due to the fact that the SR and SOs are matched on the basis of structural features only because these service description do not contain the behavioral description, such as, operation semantics and



**Fig. 2.** Reservation Scenario at HRS

<sup>5</sup> <http://www.opentravel.org>

<sup>6</sup> <http://www.harmonet.org>

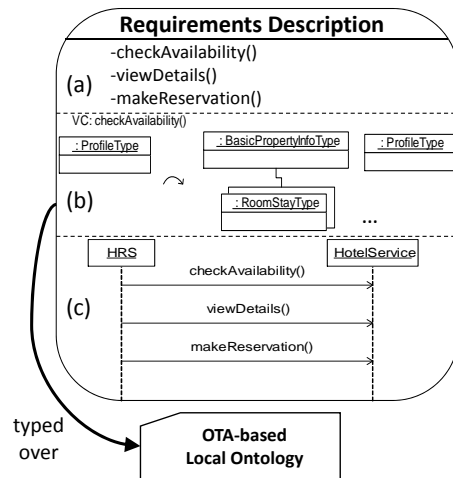


**Fig. 3.** (a) OTA-based local ontology for HRS and (b) HarmoNET-based local Ontology of HotelX and HotelY

the service protocols of the requested and provided services. A solution to overcome these limitations has to fulfill the following requirements:

- R1: A *rich service description language* is required for a comprehensive description of required and provided services containing behavioral information, such as, operation semantics and service protocols in addition to the structural information, i.e., operation signatures. The proposed language should be applicable in practical scenarios as the one described above.
- R2: A matching mechanism with heterogeneity resolution features is required for the *rich service descriptions* to enable automatic and accurate service discovery.

### 3 Rich Service Description Language



**Fig. 4.** UML-based Rich Service Description of HRS

In general, a variety of notations and languages for *rich service descriptions* already exists, such as, WSDL-S [8], Web Ontology Language for services (OWL-S) [12], and WSML[2] by Web Services Modeling Architecture (WSMX) [5], etc. However, these languages have certain limitations. For example, WSDL-S [8], which is an extension to the WSDL standard provides notations for operation syntax and semantics only and do not come up with notations for service protocols. On the other hand, languages like WSML [2] and OWL-S [12] provide concrete notations for operation syntax as well as service protocol. In case of

operation semantics, these languages do not come up with a concrete notation to specify the pre- and post-conditions of operations and leave the choice of such a rule language to the user. They are still limited to academia because these task-specific languages diverge from existing service description standards making their acceptability difficult in practice. For example, OWL-S leads to long and complex textual descriptions [14] and therefore is reported to require extra effort to learn and use.

We propose a *rich service description language* based on UML [11], which is already a de-facto industrial standard in the area of software engineering, e.g., our industrial partners HRS is already relying on UML notations and diagrams to model and describe certain service features, such as, the required workflow. Reuse of the existing UML artifacts and knowledge makes it easier for the SE community in general and service partners in our case study in particular to adapt to our proposed *rich service description* language.

We propose the following artifacts for rich service descriptions:

- (a) A description of operation signatures.
- (b) UML-based visual contracts (VC) [6, 4] for semantic description of individual operations.
- (c) UML sequence diagrams and UML statechart diagrams for requester's and provider's service protocols respectively.

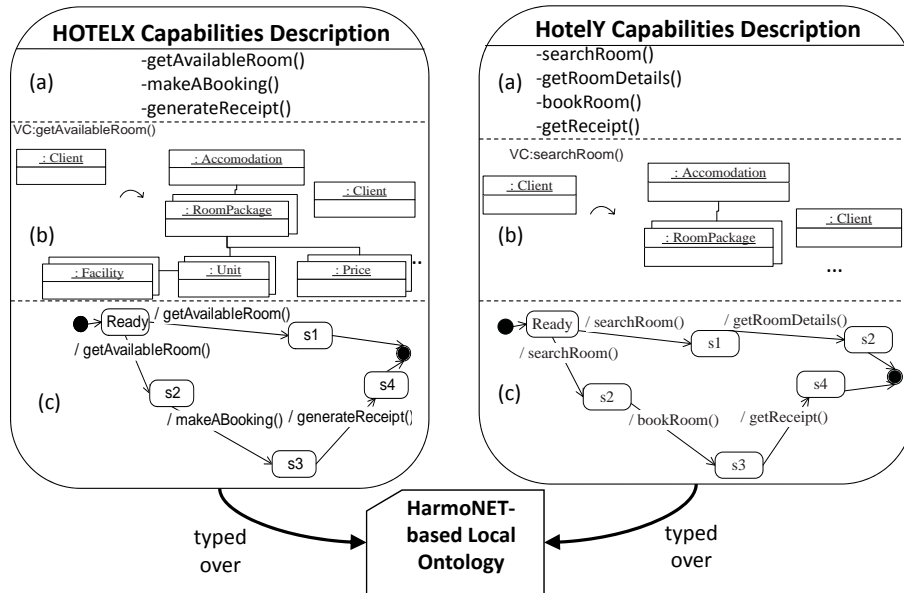


Fig. 5. UML-based Rich Service Description for HotelX and HotelY Services

The SR of HRS and SOs of *HotelX* and *HotelY* based on the proposed language are shown in Figure 4 and Figure 5.

The Web Service Description Language (WSDL) [15] as an existing standard can be used for the syntactic description of operation signatures. A VC specifies the behavior of each operation in terms of pre- and post-conditions based on UML object diagrams as shown in Figure 5. For instance, *searchRoom()* of *HotelY* requires that a *Client* object exists *before* the operation invocation and *Accommodation*, *RoomPackage* and *Unit* objects and their associations are created *after* the operation invocation.

With respect to service protocols, a requestor is interested in specifying a *desired* operation sequence invoked on a provided service. For this purpose, we use UML sequence diagrams to specify the service protocol of a requestor. However, in the case of the service provider, it is important to specify all *allowed* operation invocation sequences for a provided service. Therefore, UML statechart diagrams are the selected notation for provider's service protocol.

In the next section, we discuss the existing approaches to match service descriptions and discuss how they need to be extended to have a comprehensive matching mechanism for rich service descriptions.

## 4 Towards the Matching of Rich Service Descriptions

To enable automatic service discovery, an automatic matching mechanism for rich service descriptions is required.

In the context of our proposed language, there are already some existing VC matching approaches [6, 10]. [6] proposes a mechanism for matching an operation in the provider's SO to an operation in the requestor's SR. However, this matching mechanism has certain shortcomings. Firstly, it does not deal with the underlying heterogeneity and assumes that the service partners share a common underlying ontology. Such an assumption makes the application of this matching mechanism in practical scenarios difficult, e.g., in our example scenario, service partners conform to different ontologies from the tourism domain, i.e., OTA and HarmoNET ontologies. A potential matching mechanism has to overcome this ontological heterogeneity to match the service descriptions. Secondly, the approach is limited to 1:1 mappings between operations in the service descriptions and does not consider complex operation mappings, such as, 1:n, n:1, and n:m.

A step further in this direction is the VC matching mechanism proposed in [10]. According to this approach, a 1:n VC matching is often required in a realistic scenario, where multiple operations in provider's SO have to be invoked to fulfill the requirements specified in a single operation in requestor's SR. However, it also ignores the possibility of underlying heterogeneity of the service partner domains. Additionally, while matching, it does not consider service protocols, i.e., the intended or allowed invocation sequence of required or provided operations, respectively.

Considering these shortcomings of the existing approaches for VC matching, an elaborate and comprehensive matching mechanism is required, which enables an accurate service discovery by considering all aspects of rich service descriptions on one hand and by overcoming the underlying heterogeneity of

service partners on the other hand. In future, we aim to come up with such an automatic matching mechanism for service descriptions based on our proposed language.

## 5 Related Work

One area of related work is concerned with rich service interface descriptions. Apart from the syntactic standard WSDL [15], there are research works, such as, Visual Contracts [6], WSDL-S [8], Web Ontology Language for services (OWL-S) [12], and WSML [2] by Web Services Modeling Architecture (WSMX) [5] for semantic description in terms of their operations' pre- and post-conditions. Most of these approaches use specialized languages that are difficult to learn and use and limited in expressiveness and hence not widely used in practice yet. On the contrary, [6] is a UML-based approach [11], which is already a de-facto standard in software engineering domain.

Another important area of related work is concerned with service description matching. For this purpose, matching mechanisms [10, 7, 1] have been proposed for the service descriptions based on the languages discussed earlier. For instance, [10] proposes a matching mechanism for service descriptions based on VC-based service descriptions leaving some important issues unsolved, such as, dealing with the underlying heterogeneity, performing n:1 operation matching between service partners, and service protocol matching. Similarly, service matching approaches like [7, 1] for service descriptions based on WSML and OWL-S, respectively, do not consider service protocols while service description matching. Other approaches [9, 13, 3] also propose mechanisms for service protocol matching. However, these approaches are either not comprehensive enough in terms of interface description or ignore the underlying heterogeneity.

WSMX [5] propose a comprehensive mediator-based approach to match user goals and service capabilities for accurate service discovery while considering heterogeneity. Even though we share the same aims, our approach differs from the WSMX on the fundamental issue of using a de facto standard like UML [11].

## 6 Conclusion and Future Work

In this paper, we have identified the shortcomings of the existing service description and discovery scenario in SOC on the basis of a realistic case study of our industrial partner HRS. To overcome these shortcomings, we proposed a UML-based *rich service description language* that encompasses structural features, i.e., operation signatures as well as behavioral features, i.e., operation semantics and service protocols of requested/provided service. Such rich service descriptions comprising structural and behavioral features are a prerequisite for accurate and automatic service discovery. We briefly discussed how existing VC-based service matching approaches can be extended to define an automatic service discovery mechanism for rich service descriptions. In future, we intend to define an

automatic matching mechanism for the rich service descriptions overcoming the shortcomings of the existing service matching approaches.

## References

1. A. Mukhija, A.D.S., Rosenblum, D.S.: Dino: Dynamic and Adaptive Composition of Autonomous Services. White paper, Department of Computer Science, University College London, London (2007)
2. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language WSM: An Overview. In: Sure, Y., Domingue, J. (eds.) ESWC. LNCS, vol. 4011, pp. 590–604. Springer (2006)
3. Cavallaro, L., Nitto, E., Pradella, M.: An Automatic Approach to Enable Replacement of Conversational Services. In: Baresi, L., Chi, C.H., Suzuki, J. (eds.) ICSOC-ServiceWave '09. LNCS, vol. 5900, pp. 159–174. Springer (2009)
4. Engels, G., Güldali, B., Soltenborn, C., Wehrheim, H.: Assuring consistency of business process models and web services using visual contracts. In: AGTIVE. pp. 17–31 (2007)
5. Haller, A., Cimpian, E., Mocan, A., Oren, E., Bussler, C.: WSMX - A Semantic Service-Oriented Architecture. In: ICWS '05. pp. 321–328. IEEE Computer Society (2005)
6. Hausmann, J.H., Heckel, R., Lohmann, M.: Model-based Development of Web Service Descriptions Enabling a Precise Matching Concept. *International Journal of Web Services Research* 2(2), 67–85 (2005)
7. Klusch, M., Kaufer, F.: WSMO-MX: A hybrid Semantic Web service matchmaker. *Web Intelli. and Agent Sys.* 7, 23–42 (2009)
8. LSDIS Lab: Web Service Semantics. <http://lsdis.cs.uga.edu/projects/WSDL-S/wsdls.pdf>
9. Motahari, N., Reza, H., Benattallah, B., Martens, A., Curbera, F., Casati, F.: Semi-automated Adaptation of Service Interactions. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) WWW '07. pp. 993–1002. ACM Press (2007)
10. Naeem, M., Heckel, R., Orejas, F., Hermann, F.: Incremental Service Composition based on Partial Matching of Visual Contracts. In: Rosenblum, D.S., Taentzer, G. (eds.) FASE 2010. LNCS, vol. 6013, pp. 123–138. Springer (2010)
11. Object Management Group (OMG): Unified Modeling Language (UML) – Superstructure, Version 2.3. <http://www.omg.org/spec/UML/2.3/Infrastructure> (2009)
12. OWL-S Coalition: OWL-based Web Service Ontology. <http://www.ai.sri.com/daml/services/owl-s/1.2/> (2006)
13. Poizat, R.M.P., Salaün, G.: Adaptation of Service Protocols Using Process Algebra and On-the-Fly Reduction Techniques. In: Bouguettaya, A., Krüger, I., Margaria, T. (eds.) ICSOC'08. LNCS, vol. 5364, pp. 84–99. Springer (2008)
14. Sabou, M., Richards, D., van Splunter, S.: An experience report on using daml-s (2003)
15. W3C: Web Service Description Language(WSDL). <http://www.w3.org/TR/wsd120/> (2007)