Enabling Semantic Mediation in DaaS Composition: Service-Based and Context-Driven Approach

Idir Amine Amarouche, University of Sciences and Technology Houari Boumediene, Bab-Ezzouar, Algeria

Djamal Benslimane, Claude Bernard Lyon 1 University, Lyon, France

Zaia Alimazighi, University of Sciences and Technology Houari Boumediene, Bab-Ezzouar, Algeria

ABSTRACT

As commonly agreed, Web services fall into two categories depending on their functionality world-altering services and Data-as-a-Service (DaaS). Much work has been done on automatic DaaS discovery and composition, such as the query rewriting approach proposed by the database community. In this context, DaaS is described as Parameterized-RDF View over Domain Ontology (DO). However, the DO is unable to capture the different perspectives or viewpoints for the same domain knowledge. This limitation raises semantic conflicts between pieces of data exchanged during DaaS composition process. Thus, mediators are typically required to reconcile potential conflicts. In this paper, the authors propose a service-based approach for automatically inserting appropriate mediation services in DaaS compositions to resolve incompatibilities in their data flow. Also, the authors present a context-driven approach to support semantic mediation between composed DaaSs. The implementation and the experimental evaluations performed showed us satisfactory results.

Keywords: Conflict, Context, Data-As-A-Service (Daas), DaaS Composition, Mediation Service, Parameterized RDF View, Query Rewriting, RDFS, Semantic SPARQL Query

INTRODUCTION

An important class of web services consists of those that primarily do data processing and produce output data (Saleh et al., 2009; Truong & Dustdar, 2009). This type of Web service is known as Data-as-a-Service. DaaS services return collections of Data for a given set of parameters without any side effects (Carey, 2006). DaaS services composition is a powerful mean to answer users’ complex queries.

Several approaches are proposed to enable automatic Web service composition (Rao & Su, 2005). The Semantic-based ones proceed by describing the Web services properties over ontology. In fact, many ontology languages like OWL-S¹ or WSMO² and extension mechanisms like WSDL-S (Akkirajuet al., 2005) or

DOI: 10.4018/ijitwe.2013100101
SAWSDL\(^3\) provide standard means by which WSDL (Web Service Description Language) document can be related to semantic description. However, these solutions do not provide a way to relate semantically the Web service parameters (i.e. input and output) which hampers their applicability to DaaS composition. The automation of DaaS composition requires the specification of the semantic relationships between inputs and outputs parameters in a declarative way.

This requirement can be achieved by describing DaaS as view over a Domain Ontology (DO) following the mediator-based approach (Wiederhold, 1992). In this context, several works (Barhamdji et al., 2010; Zhou et al., 2008; Vaculin et al., 2008) consider DaaS as Parameterized RDF View (PRV) with binding pattern over a DO. The PRV describes how the input parameters of the DaaS relate to the data it provides. Defined views are then used to annotate DaaSs description files (e.g. WSDL files) and are exploited to automatically compose DaaSs. Thereby, the DaaS composition problem is reduced to a query rewriting problem largely studied in data integration field (Halevy, 2001).

However, there are several references ontologies which formalize the same domain knowledge. Thus, the construction of a DO, unifying all existing representations of real-world entities in the domain, is a strong limitation to interoperability between DaaSs.

This limitation raises semantic conflicts between pieces of data exchanged during DaaS composition. To this end, the applicability of the previously cited DaaS composition approaches is not practical.

Therefore, considering the semantic conflict detection and resolution during the composition process is crucial.

**Motivating Example**

We provide an illustrating example where the information needs of health actors are satisfied with a DaaS Composition System (DCS). The DCS is inspired from the framework proposed by Barhamdji (2010). We will consider an e-health system which exports a set of DaaSs to query patient data as presented in Table 1. The inputs and outputs of DaaS parameters are denoted respectively by the symbols ‘$’ and ‘?’.

We assume that a physician submits the following query \(Q\): ‘what are the states indicated by the recent Blood Pressure Readings (BPR) for a given patient’’. We assume that the DCS will automatically generates DaaS composition (i.e. including the following DaaS related tasks: discovery, selection and composition), as response to physician query, including respectively \(S_1\), \(S_2\) and \(S_3\) as depicted in Figure 1(a). The DCS invokes automatically in the following order:

- “\(S_1\)” that provides the recent Vital Sign Exam performed on a given patient;
- “\(S_2\)” to retrieve the BPR from a given Vital Sign Exam. The BPR is represented by two concatenated values. For instance, BPR is “120/80 cm/HG” where 120 is the Diastolic BPR (BPR.D) value, 80 is the...

### Table 1. Example of DaaS services

<table>
<thead>
<tr>
<th>DaaS</th>
<th>Service functionality and context</th>
</tr>
</thead>
</table>
| \(S_1(x, ?y)\) | Returns recent Vital Sign Exam “\(y\)” performed on patient “\(x\)”.
| \(S_2(x, ?y)\) | Returns Blood Pressure Reading (BPR) “\(y\)” mentioned in given Vital Sign Exam “\(x\)”.
  **Context:** “\(y.value\)” is represented by two concatenated values (BPR Diastolic/ BPR Systolic) having “mm/HG” as unit of measurement; “\(y.code\)” is the exam code expressed in LOINC.
| \(S_3(x, ?y)\) | Returns the BPR state “\(y\)” of the Mean Arterial Pressure (MAP) “\(x\)”.
  **Context:** - “\(x.value\)” is the MAP value having “cm/HG” as measurement unit; - “\(x.code\)” is the exam code expressed in SNOMED; - “\(y.value\)” is the BPR state value represented according to the new classification code (e.g. stage 1, 2, 3, 4).
Image Mosaicing Using Binary Edge Detection Algorithm in a Cloud-Computing Environment
www.igi-global.com/article/image-mosaicing-using-binary-edge-detection-algorithm-in-a-cloud-computing-environment/164468?camid=4v1a

Access to Device Reachability Data as a Component of Horizontal M2M Service Platform
www.igi-global.com/article/access-to-device-reachability-data-as-a-component-of-horizontal-m2m-service-platform/193007?camid=4v1a