268 *Int. J. Web and Grid Services, Vol. 1, Nos. 3/4, 2005*

Towards a semantic- and context-based approach for composing web services

Michaël Mrissa* and Djamal Benslimane

Laboratoire LIRIS, Bâtiment Nautibus
Université Claude Bernard Lyon 1
69622 Villeurbanne CEDEX, France
E-mail: michael.mrissa@liris.cnrs.fr
E-mail: djamal.benslimane@liris.cnrs.fr
*Corresponding author

Zakaria Maamar

Zayed University
College of Information Systems
P.O. Box 19282, Dubai, UAE
E-mail: zakaria.maamar@zu.ac.ae

Chirine Ghedira

Laboratoire LIRIS, Bâtiment Nautibus
Université Claude Bernard Lyon 1
69622 Villeurbanne CEDEX, France
E-mail: chirine.ghedira@liris.cnrs.fr

Abstract: Despite the widespread adoption of web services, several obstacles still hinder their smooth automatic composition. First, techniques that exploit semantic information during web services discovery are still lagging behind despite multiple initiatives like OWL-S. Second, the context in which web services evolve is, to a certain extent, ignored. This prevents deploying adaptable web services. In this paper, we propose a semantic- and context-based approach for web services composition. By semantics, we mean the capacity of web services, which are engaged in interactions, to understand what these interactions are about and how to carry out these interactions. By context, we mean the capacity of web services to assess their current capabilities and ongoing commitments before these services participate in any composition.

Keywords: web services; composition; context; semantics; ontology.

Reference to this paper should be made as follows: Mrissa, M., Benslimane, D., Maamar, Z. and Ghedira, C. (2005) 'Towards a semantic- and context-based approach for composing web services', *Int. J. Web and Grid Services*, Vol. 1, Nos. 3/4, pp.268–286.

Biographical notes: Michaël Mrissa is a PhD student at Claude Bernard University, Lyon, France. He is a member of the LIRIS-CNRS Research Laboratory. His research interests include web services composition and mediation, the semantic web and P2P computing.

Djamal Benslimane is a Full Professor at Claude Bernard Lyon 1 University, Lyon, France. He is a member of the LIRIS-CNRS Research Laboratory. He obtained his PhD from Blaise Pascal University, Clermont-Ferrand, France, in 1992.

Zakaria Maamar is Associate Professor of Computer Science at Zayed University, Dubai, United Arab Emirates. His research interests include web services, software agents and context-aware computing. Maamar has a PhD in Computer Science from Laval University in 1998.

Chirine Ghedira is Associate Professor at Claude Bernard Lyon 1 University, Lyon, France. She is a member of the LIRIS-CNRS Research Laboratory. She obtained her PhD in Computer Science from the National Institute for Applied Sciences (INSA) of Lyon, France, in 2002.

1 Introduction

1.1 Motivation

Web services are backing the emergence of a new type of systems that tend to cross-cut companies' boundaries. A web service is a software component that other software components and humans can discover and trigger to satisfy their needs (*e.g.*, hotel booking). Several standards are associated with web services for their discovery, description and binding (Milanovic and Malek, 2004). Taking advantage of the eXtensible Markup Language (XML), web services also have the capacity to be composed into high level business-processes usually referred to as composite services. Composition primarily addresses the situation of a user request that cannot be satisfied by any available web service, whereas a composite service obtained by integrating available web services might be used (Berardi *et al.*, 2003). Composing services rather than accessing a single service is essential. It offers better benefits to users. For instance, a web service in charge of planning outdoor activities needs to consider checking the weather forecasts, which another web service offers, before making any proposal for such activities. The collaboration between both web services is deemed appropriate. For composition requirements, a composite service is always associated with a specification that describes, among others, the list of component web services that participate in the composite service, the execution order of the component web services and the corrective strategies in case of exceptions. There exist multiple languages for the specification of web services compositions including the Web Services Flow Language (WSFL) (Leymann, 2001) and the Business Process Execution Language (BPEL) (Curbera *et al.*, 2003).

The specification of composite services is also concerned with the semantics of information that component web services exchange (Medjahed *et al.*, 2003). The need for a common semantics is intensified when web services, which originate from distinct providers, participate in the same composition. Despite tremendous efforts in web services composition, very little has been accomplished so far regarding the semantic reconciliation of web services. Several obstacles still hinder the automation of semantic mediation. First, techniques based on semantic information for web services composition are still lagging behind despite the pressing needs of users to achieve application integration. Semantic descriptive languages for web services (*e.g.*, OWL-S: Ontology Web Language-based Web Service Ontology) are among the initiatives that help boost semantic use during the various steps featuring service composition such as discovery, mediation and monitoring. Second, the context in which web services evolve is, to a certain extent, ignored. This prevents deploying adaptable web services. For a web service, being context aware means the ability of detecting and responding appropriately to changes in the environment (Maamar *et al.*, 2005a; Rios *et al.*, 2003).

1.2 Challenges

Despite the widespread use of web services, these services still lack the capabilities that could propel them to the acceptance level of traditional integration middleware (*e.g.*, CORBA, RMI Jini). This lack is somehow due to the trigger-response pattern that is imposed on the interaction of web services with the external environment. The compliance with the trigger-response interaction pattern means that a web service only processes the requests (*e.g.*, SOAP over HTTP-based) that it receives without considering its internal status in terms of execution, commitments and availabilities. However, there exist several situations that call for the self-management of web services so that the challenges of scalability, flexibility, stability and autonomy are properly faced. By scalability, we mean the capacity of a web service to interact with a small or large community of web services without having their expected performance either disrupted or reduced. By flexibility, we mean the capacity of a web service to adapt its behaviour according to the situation in which it operates. By stability, we mean the capacity of a web service for resisting unforeseen changes while maintaining operation and recovering to normal levels of operation after disturbances. Finally by autonomy, we mean the capacity of a web service to accept requests of participation in composite services or reject such demands in case of unappealing rewards.

While scalability, flexibility, stability and autonomy challenges put developers of web services under the pressure of satisfying businesses' promises of delivering web services-based solutions, information heterogeneity is another one that needs to be looked carefully into. Because independent providers develop web services, achieving their semantic composition is another major step to carry out. Matching inputs/outputs during web services composition is by far not sufficient. Semantic alteration or reduction could negatively affect the quality of the exchanged information between web services.

1.3 Objectives

The use of semantics is a cornerstone to the composition of web services. Semantics that enhance web services could be classified into four types (Sheth and Ramakrishnan, 2003):

1. functional semantics that describes their action
2. QoS semantics that gives details about their characteristics like response time, cost, reliability and fidelity
3. data semantics that describes their input and output interfaces
4. execution semantics that encompasses the idea of message sequence, conversation pattern, flow of actions, preconditions and effects of web services invocation.

In this paper, we focus on the semantics for solving information discrepancies (third type) and the actions that permit an automated semantic reconciliation of web services.

Besides the semantics issue, further issues still exist during web services composition such as which businesses have the capacity to provide web services, when and where the provisioning of web services occur and how web services from independent providers coordinate their activities so that conflicts are avoided. To deal with some of these issues, it was recommended considering the context in which web services composition and execution occur (Maamar *et al.*, 2005a). Context is the information that characterises the interaction between humans, applications and surrounding environment. In this paper, we also focus on assessing the value-added of context to web services composition. Therefore, our objective in this paper is to define an approach that enables semantic interactions between context-aware web services. In addition, it should be noted that contextualising ontologies is outside the scope of this paper. Interested readers are referred to Benslimane *et al.* (2003) and Bouquet *et al.* (2004).

Organisation of the paper

Section 1 provides an overview of the paper in terms of challenges to expect during web services composition and solutions to adopt for these challenges. For understanding purposes, Section 2 describes some concepts. Section 3 briefly presents a motivating example that will be used throughout this paper. Section 4 is about our proposed approach for integrating semantics and context into web services composition. Section 5 details the way web services are made context aware. Section 6 overviews some related work. Finally, Section 7 concludes the paper.

2 Background

Current approaches only achieve web services composition at the level of message interactions (Maamar *et al.*, 2005b). This is by far not sufficient because composition also needs to be conducted at the level of message semantics. The objective of semantic composition is to ensure that web services understand the information they exchange. The need for a common semantics is intensified when web services, which originate from different providers, take part in the same composition. Thus, web services have to be initially checked whether they can work together or not (Medjahed *et al.*, 2003). To tackle the semantic obstacle, web services should bind to appropriate ontologies according to the situations in which they participate. A situation corresponds to the application domain of a web services composition such as vacation planning.

2.1 What is a web service?

A web service is a software component that is as independent as possible from specific platforms and computing paradigms. It is mainly developed for inter-organisational cases. Also, it is easily composable so that developing adapters during composition is not required (Benatallah *et al.*, 2003).

2.2 What is ontology?

Gruber (2002) defines ontology as "a logical theory accounting for the intended meaning of a formal vocabulary, more specifically, its ontological commitment to a particular conceptualisation of the world". In this paper, we comply with the definition that ontology is 'a logical theory' that describes the meaning of a formal vocabulary, from a particular 'conceptualisation of the world' (referred to as 'point of view' in the following). This definition assumes that different ontologies can be heterogeneous and exhibit various discrepancies.

2.3 What is context?

Dey defines context as any information that can be used to characterise the situation of an entity (person, place or object) that is considered relevant to the interaction between a user and an application (Dey *et al.*, 2001). This information can be about the circumstances, objects or conditions by which the user is surrounded. Many researchers have attempted defining context. Among these researchers, Schilit *et al.* (1994) propose three categories of context: computing category (*e.g.*, communication cost, bandwidth), user category (*e.g.*, nearby people, social situation), physical category (*e.g.*, traffic conditions, temperature).

3 Motivating examples on semantic mediation

3.1 The weather forecasts scenario

Let us assume a web service that provides weather information on a given place, and a second web service that forecasts the weather for the next five days, being given a particular place and some information about the current weather. Therefore, 'getCurrentWeather' and 'getWeatherForecast' web services need to collaborate so that 'getCurrentWeather' feeds 'getWeatherForecast' with relevant data (Figure 1).

Web service description is usually split into two levels. The first level, an abstract level, describes the web service's operations. The second level, a grounding level, describes the web service's exchange protocols and data encoding. However, semantics is totally absent from both descriptions. The semantics of input and output interfaces may differ from one web service to another. In a similar case, mappings are needed between output data of the first web service and input data of the second web service. Thus, a data free-of-conflict composition is enabled (Medjahed *et al.*, 2003).

In the beginning of this paper, several types of heterogeneities related to web services composition were listed. In the following, and based on previous work (Mrissa *et al.*, 2005), we refine them and illustrate with the discrepancies between the data structures of 'getCurrentWeather' and 'getWeatherForecast' web services.

1 Label conflict

   Different names are given to parameters. Indeed, 'Temp' parameter in 'getCurrentWeather' corresponds to 'Temperature' parameter in 'getWeatherForecast'. This is true even though a unit conflict exists between them. A typical solution uses translation rules that are stored in a library and fired upon request.

2 Merging conflict

In ‘getCurrentWeather’, ‘City’ and ‘Country’ parameters are dealt with separately. However, in ‘getWeatherForecast’ both parameters are combined using the ‘Place’ parameter. This calls for merging both parameters into a single one so that the input requirements of ‘getCurrentWeather’ are satisfied. ‘City’ and ‘Country’ parameters of the first web service will correspond to ‘Place.City’ and ‘Place.Country’ in the second web service, respectively.

3 Data representation mismatching

‘Pressure’ parameter of ‘getCurrentWeather’ is not relevant to ‘getWeatherForecast’. Thus, it could be dropped during composition.

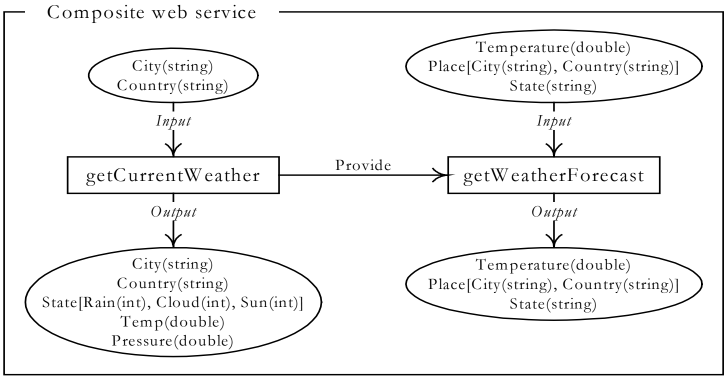
4 Unit conflict

Temperature is expressed in Fahrenheit and Celsius degrees in ‘getCurrentWeather’ and ‘getWeatherForecast’, respectively. A conversion function is needed between both units.

5 Values interpretation conflict

For illustration, ‘getCurrentWeather’ returns three values (‘rain’, ‘clouds’ and ‘sun’, which are restricted between 0 and 100) for describing the respective percentages of rain, clouds and sun in the sky. However, ‘getWeatherForecast’ expresses the sky’s state with a single value of type string (‘sun’, ‘rain’ or ‘clouds’). A conversion function is needed so that the structures of both web services are adapted. This function provides a string value to ‘getWeatherForecast’ after selecting the highest value among ‘rain’, ‘clouds’ and ‘sun’.

Figure 1 Sample of a composite web service



3.2 Motivation for an ontology-based mediation

A straightforward solution to semantic heterogeneity is to develop a mediation code among a set of web services. Such mediation techniques were already introduced in the domain of databases (Wiederhold, 1992). Once the accurate mediation code is available, it is fairly simple to call it when necessary. However, a hard-coded mediation is only limited to a specific set of web services. It is not reusable with other sets of web services. To address this limitation, two features need to be enabled for a scalable and adaptable mediation between web services.

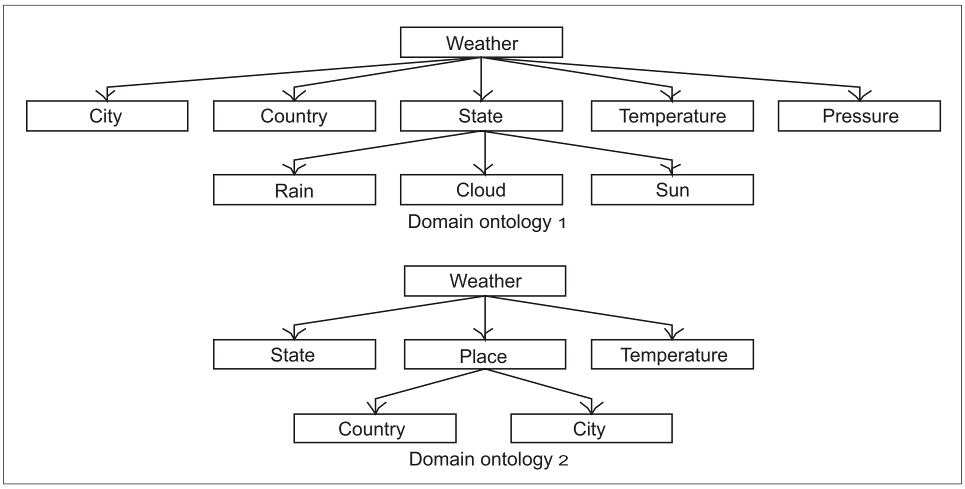
First, we deem appropriate using domain ontologies associated with large numbers of web services (Rios *et al.*, 2003). We consider ontologies as reliable references that are made available on the internet and that can be accessed from their Uniform Resource Identifiers (URIs). Each provider may decide to build and post its ontology. Doing so enables semantic interoperability with all the associated web services. As a further step in the process of enabling inter-ontology communications for web services composition, the use of a shared ontology, designed as a common description for an application domain, is also deemed appropriate.

Second, we back the rationale of an active mediation component for supporting automated and semantic-aware mediation, which is at least partially automated. Accessing providers' ontologies and conversion functions enables the automation of a part of the mediation process between web services. Mediation components use ontologies to take advantage of semantic descriptions and to determine the best way for solving data discrepancies. Also, a storage means needs to be designed for conversion functions such as function libraries. The design of generic and parameterised sets of functions is also to be considered. Thus, conversions possibilities are scalable over new systems to be integrated.

3.3 Ontology mappings

Our solution for a free-of-conflict web services composition is built upon the generation of mappings from the domain ontologies of providers to a shared ontology (Mena *et al.*, 2000; Sheth *et al.*, 2003). The shared ontology is designed with the specific purpose of acting as reference ontology for all the forthcoming web services compositions in a specific application domain (in this paper, weather statement and forecasts). Figure 2 presents two domain ontologies based on the example of Section 3.1. These ontologies have possibly been designed by independent providers and thus are heterogeneous. Samples of heterogeneities are similar to the ones presented between 'getCurrentWeather' (left side of Figure 2) and 'getWeatherForecast' (right side of Figure 2) web services.

Figure 2 Domain ontologies per provider



The creation of mappings from the first-domain ontology to the second-domain ontology and *vice versa* enables the web services of each domain to communicate with each other. Let us assume that a mediation component is in charge of supervising the communication between these two web services, and that the mappings between the ontologies are stored and available to the component, for instance in a library. Then, the mediation component uses these functions so that a direct communication between web services is achieved during composition. Once the set of functions is made available to the mediation component, it is possible at execution time to transform the data exchanged between web services. The process of creating conversion functions between two ontologies is only executed once. This saves time for further composition executions. However, this solution only concerns a pair of ontologies. If *n* ontologies had to exist, the design of $n*(n-1)/2$ mappings would be required, which would not be wise to adopt.

To overcome the limitations of developing mappings between ontologies, adopting a shared ontology is promising (Mena *et al.*, 2000; Sheth *et al.*, 2003). A shared ontology is a bridge between the ontologies of domains. The motivation for a shared ontology is the storage of the mappings from local ontologies. When they are stored, they can be reused next time the mediation component needs them. This solution saves time and energy when it comes to composing web services that bind to local ontologies. The mediation component accesses the mapping storage point. Then, it dynamically converts the information from the local ontology of web services to the shared ontology, and the other way round. Therefore, the use of the shared ontology significantly reduces the need for building direct mappings between local ontologies that were not planned to be together. The sets of conversion functions to be used during mediation must be carefully designed for scalability reasons. Table 1 presents a set of generic functions that could provide support for conversion between the local ontologies of Figure 2 and a shared ontology. Thus, reusing in different composition scenarios is enabled.

Table 1 Some potential conversion functions

Return value	*Name*	*Parameters*	*Description*
Object	changeLabel	object element string newLabel	Changes the label of element.
Double	changeUnit	string unit1 string unit2 double value	Converts the value entered from a unit system to another.[1]
Object	mergeObjects	object [ ] tab	Merges different elements into a single object.
String	greatestLabel	integer [ ] val	Compares the values of the table and returns the label of the greatest.

Note: 1 Example: unitChange (Celsius, Fahrenheit, 20) converts the value 20 from Celsius degrees into Fahrenheit

4 A semantic- and context-based approach for web services composition

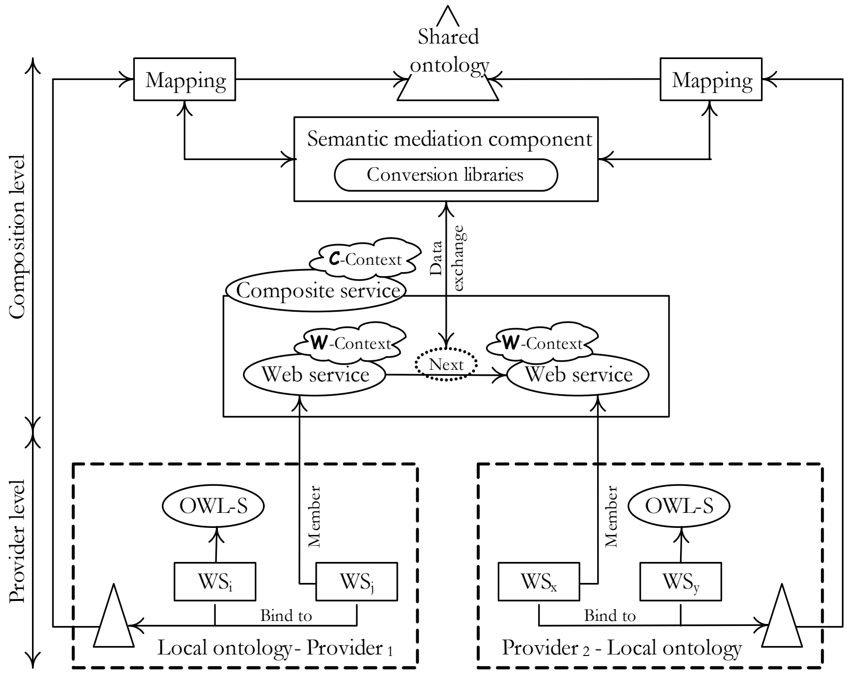
4.1 Architecture

Figure 3 depicts the architecture of our proposed approach for web services composition. This architecture enables meeting the objectives of developing context-aware web services that are engaged in free-of-conflict semantic interactions. The architecture encompasses multiple components located at two different levels: the provider and the composition. The provider level identifies the providers of web services, who use different local ontologies to define their web services (*e.g.*, WSDL). The composition level provides the relevant functionalities for supporting the development of composite services. This level includes five elements:

1. A shared ontology that contains concepts on which web services (providers) have already agreed for the needs of mediation.
2. Ontology mappings that describe the conversions between each local ontology and the shared ontology.
3. Mediation components that use (to a certain extent) these mappings during the execution of composite services.
4. Conversion libraries that provide multiple conversion functions and support the mediation component in their operations.
5. A context structure that is associated with web services and composite services:
    - *The shared ontology* captures the semantics of the concepts of an application domain. The ontology defines a vocabulary that allows a consistent exchange among web services. Different languages are proposed for expressing ontologies. In our approach, OWL is adopted (Dean *et al.*, 2002), which permits to detect semantic similarities among local ontologies and to convert values from one web service into another. The designer of the architecture presented in Figure 3 must choose or create the shared ontology that the semantic mediation component refers to. It is a particularly arduous task. This is because the designer has to consider and choose between different alternatives of shared ontology. Also, these choices will directly determine the quality of mediation between providers' ontologies.
    - *Ontological mappings* implement the relationships between the terms of a local ontology and their related terms in the shared ontology. These relationships are expressed as built-in axioms that OWL provides. The semantics that is used during ontological mappings helps the mediation component (discussed below) to determine the appropriate conversion steps that translate data from a web service to another in the composition.
    - *Local web services* are semantically described using OWL-S representations that are obtained out of their standard WSDL descriptions enhanced with semantic features (Martin *et al.*, 2004). Each provider develops an OWL-S description using its local ontology for the web services it provides.

- *Semantic mediation component* sets mappings between heterogeneous OWL-S descriptions. When a web service invokes a peer, the mediation component uses ontological mappings to convert data output from the first web service to a structure that meets the shared ontology's expectations. Then, the mediation component converts the data from the shared ontology into the local ontology of the second web service.
- *Conversion functions libraries* store generic conversion functions that the mediation component triggers when needed. These functions are designed to be generic. Thus, a large coverage is enabled of the possible conversions between data formats and representations. When the mediation component interprets the ontological mappings, it accesses the conversion libraries that help achieve the data conversion from a format to another. A small database containing several data formats supported by the functions may provide a reliable back-end to the libraries.

Figure 3 Architecture for semantic- and context-based approach for web services composition



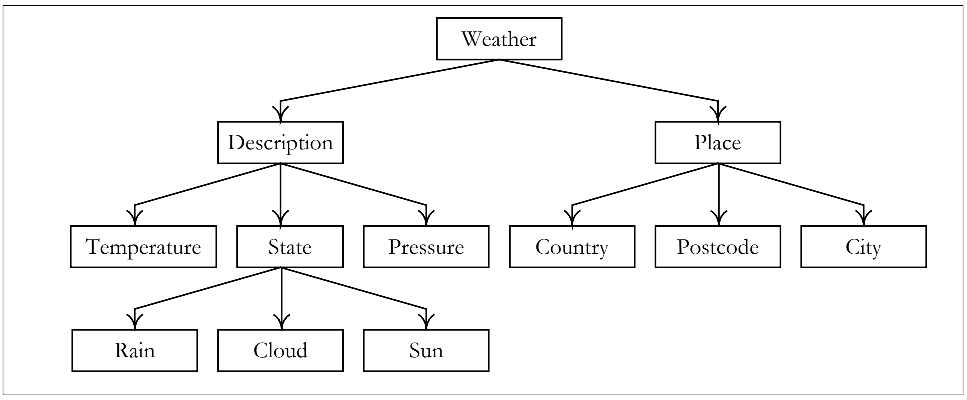
Besides the heterogeneity of local ontologies, additional issues/challenges exist during the composition of web services. These issues/challenges include which businesses have the capacity to provision web services, when and where the provisioning of web services occurs and how web services from independent providers coordinate their activities so that conflicts are avoided. To deal with these, a context structure is considered for each web or composite service. Additional details on the rationale of associating services with context are given in Maamar *et al.* (2005a).

Because web services can participate in different compositions, a structure is required for tracking their participation in each composition. This structure, which is referred to as the W-context in Figure 3, provides information about the status of a web service (*e.g.*, number of active participations, execution progress per active participation). In addition, to monitor and assess the deployment of the specification of a composite service, another structure, which is referred to as the C-context in Figure 3, is also required which includes various information related for example to the identification of the previous/current/next web services of a composite service and their respective execution status.

4.2 Ontologies, ontological mappings, and mediation

In most composition scenarios, web services originate from different providers. Each provider uses a distinct ontology to describe the interfaces of its web services. Providers have different ways of modelling the knowledge of a domain. To address some of the semantic composition challenges, we promote the use of a shared domain ontology. This ontology is intended to encompass as much as possible the different ontologies that could be designed for describing the domain. It is highly recommended that the shared ontology extends the ontologies of the concerned providers. Each interested third party could adapt its domain ontology or even create the mappings from its domain ontology for better inter-operability within the shared ontology. Figure 4 illustrates an example of a shared ontology specifically designed for supporting the mappings between the domain ontologies of Figure 2.

Figure 4 Shared domain ontology



In Figure 3, the architecture assumes that the composition designer uses the existing mappings between the providers’ domain ontologies and the shared ontology. This saves time to the composition designer because the mappings for all the web services of a domain are generated only once. Once generated, mappings between a provider’s domain ontology and the shared ontology are used during execution so that the data exchanged between web services are converted.

Supplying OWL descriptions of web services is not enough to support data conversion between web services. Another issue arises – how to express and store the mappings. Simple mappings between classes of the ontologies can be expressed as simple OWL relations (inclusion, equivalence, *etc.*). However, there are still some concerns about storing complex conversion functions. The use of an inference engine generating complex translation rules backed with a set of primary translation functions, as reported in Spencer and Liu (2004), is an alternative to consider. Spencer and Liu proposed that a complex conversion can be decomposed into basic conversions that are used together as a concatenation of conversions. A rule-based inference engine uses a set of basic conversions that are considered as rules. This engine reasons over the web services descriptions and generates multiple data transformation rules. At execution time, data exchanged between web services are stored in an inference queue. The accurate transformation rules for obtaining the data structure required by the next web service are applied.

Recent techniques for mediation between web services have adopted automatic mappings. For instance, Patil and Oundhakar have developed the Semantic Annotation of Web Services (SAWS) algorithm. SAWS consists of automatically mapping each concept defined in a WSDL description onto an ontological concept (Patil and Oundhakar, 2002). In particular, the SAWS algorithm compares a concept in WSDL to a concept of the ontology in order to return the degree of similarity (using a scale of 0 to 1) between them. The comparison occurs at the element and structure levels. At the element level, the algorithm calculates the linguistic similarity between the concepts. At the structure level, the algorithm calculates the similarity between the concepts' subtrees.

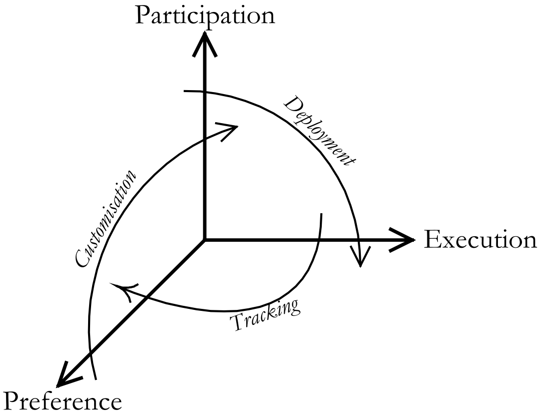
Besides the SAWS algorithm, other algorithms to perform mappings between two schemas or ontologies exist. Cupid, an algorithm described in Madhavan *et al.* (2001), identifies the potential mappings between schema elements based on their names, data types, constraints and schema structures. The problems that may arise here relate to mismatching between ontologies or schema and/or ontology versioning. There also exists another algorithm, which is reported in Patil *et al.* (2004), that permits the conversion of models in XML schema and ontology into a common representation format called SchemaGraph. It is a set of nodes connected by edges and provides a generic solution to ensure the mutual mapping between XML schema and ontology, wherein ontologies in DAML, RDF-S, OWL or other languages. Conversion functions are needed to convert both XML schema and ontology into SchemaGraphs. Every concept from the WSDL SchemaGraph is compared to concepts from the ontology SchemaGraph. For that process of comparing, a special function (findMapping) returns first the mapping between a WSDL and ontology concept pair and, second, a matching score.

To wrap up this section, the best alternative for conducting mappings consists of converting each web service description into a schema graph. Then, one of the aforementioned techniques to generate mappings between the shared ontology and the descriptions of web services is applied. These mappings will be saved on a mapping registry for further use. Thus, to compose web services, the appropriate conversion functions are called upon request in order to achieve the necessary mediation among those web services.

4.3 Contextualisation of web services

Context-aware computing refers to the ability of a software application to detect and respond to changes in its environment. Making web services context aware is not straightforward. Many issues need to be addressed (adopted from Satyanarayanan, 2001): how is context structured, how does a web service bind to context, where is context stored, how frequently does a web service consult context, how are changes detected and assessed for context update purposes and what is the overload on a web service for taking context into account. In order to deal with these issues, the context of a web service needs to be organised along three interconnected perspectives (Figure 5). The participation perspective is about overseeing the multiple composition scenarios in which a web service concurrently takes part. This guarantees that the web service is properly specified and is ready for execution in each composition scenario. The execution perspective is about looking for the computing resources on which a web service operates. It is also about monitoring the capabilities of these computing resources so that the web service's requirements are constantly satisfied. It has been argued that the availability of the computing resources affects the capabilities of a web service (Maamar *et al.*, 2004a). Finally, the preference perspective is about ensuring that user preferences of type execution time (*e.g.*, at 2 pm) and execution location (*e.g.*, user passing by meeting room) are integrated into the specification of a composite service.

Figure 5 Context organisation of a web service

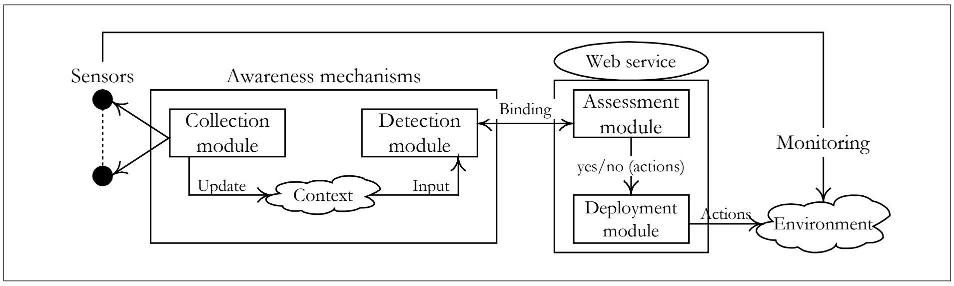


In Figure 5, participation, execution and preference perspectives are interconnected. First, deployment connects participation and execution perspectives. It also highlights the web service that is executed once it accepts participating in a composition. Second, tracking connects execution and preference perspectives. Tracking highlights the significance of monitoring the execution of a web service so that user preferences are properly handled. Finally, customisation connects preference and participation perspectives. Customisation highlights the possibility of adjusting a web service so that it can accommodate various user preferences. The integration of context into web services composition ensures that the requirements of and constraints over these web services are taken into account. While current composition approaches rely on different selection criteria (*e.g.*, execution cost and reliability), context supports web services in their decision-making process when it comes to whether accepting or rejecting participation in a composition.

5 Making web services context aware

By essence, web services are not aware of their surrounding environment. The use of awareness mechanisms, as suggested in Figure 6, allows web services to assess the environment they have to deal with, which is referred to as context, and to adapt the deployment execution in consequence. To make web services context aware, developing collection, detection, assessment and deployment modules is called for.

Figure 6 Connection between web service and context



The collection module gathers raw information from several sensors. It then updates the context information repository. Here arises an important problem relative to context information heterogeneity. Our solution to contextual information heterogeneity refers to the work presented in Power *et al.* (2004), which proposes the use of an ontology mapping repository for mapping context information. The work of Power *et al.* is related to P2P networks, in which some specialised peers store mappings between context ontologies. In our architecture, it is the collection module that stores context ontology mappings. We are working towards automatic mappings generation, as presented in Spencer and Liu (2004), but applied to context information. The objective is to convert raw data from sensors to structured data and to map these data onto the most accurate correspondence that can be found with our context ontologies stored in the ontology mapping repository. The detection module analyses current context data in order to communicate context changes to the assessment module. It represents a link between the awareness mechanism and the web service deployment building blocks. The assessment module is a logical structure that infers sets of rules from the information it receives from the detection module. The assessment module also controls the deployment module of the web service itself. The deployment module is in charge of executing the necessary actions for deploying the web service.

In Maamar *et al.* (2005b), a context ontology was developed and was called OWL-C. It enabled the specification of the internal structure of context according to the type of service, whether web or composite. From a web services perspective, context is defined as a set of common metadata about the current execution status of a web service and its capability of collaborating with peers, possibly enacted by distinct providers. In

Maamar *et al.* (2005b), a set of arguments that constitute the structure of context was suggested per type of service (web or composite). Table 2 suggests some arguments that are associated with the W-context of a web service (Figure 3). With regard to the C-context of a composite service, some potential arguments could be (Maamar *et al.*, 2005b): previous web services, current web services, next web services and status per web service instance.

Table 2 Some arguments of W-context

Number of participants allowed	*Number of active participations*	*Status/composite service*
Corresponds to the maximum number of compositions in which a web service can participate	Corresponds to the number of compositions in which the web service is currently participating	Corresponds to the status of the web service per composition

OWL-S organises the description of a web service along three categories: profile, process model and grounding. OWL-C also guarantees that the description of a service context happens along the same categories. The profile describes the arguments and capabilities of context (*e.g.*, What does context require and provide?). The process model suggests how context collects raw data from sensors and detects changes that need to be submitted to the service. Finally, the grounding defines the bindings (protocol, input/output messages, *etc.*) that make context accessible to a service. In addition, because context is an argument of the structure of a service, the context profile can be used during the selection of web services for composition. It is interesting to note that altering contextual information during its transfer between web services and composite services has negative consequences on the normal progress of a composition. Some consequences could be the adoption of the wrong strategy for selecting a component service (*e.g.*, favouring execution-cost criterion over reliability criterion), or poorly assessing the exact execution status of a service (*i.e.*, service being suspended while it was assumed under execution).

6 Related work

Semantics and context are intensively explored for the needs of a new generation of information systems (*e.g.*, mobile information systems, web information systems). In this section, we only consider the initiatives that combine semantics, context and web services. This related work is seen along two perspectives: semantics for web services and context for web services.

6.1 Semantics for web services

A great deal of work has been done on semantic web services. The semantic description of web services lets intelligent components (*e.g.*, software agents) understand functionalities of such web services. This allows them to be discovered and invoked. Two main languages, OWL-S and WSMO (Web Service Modelling Ontology) (Arroyo and Stollberg, 2004), are proposed to describe web services semantically. While OWL-S is based on the combination of OWL and WSDL, WSMO uses F-Logic and XML-based features of web services. Based on one of the previous languages, several research projects have been developed on semantic web services (*e.g.*, ODESWS, METEOR (LSDIS), and SWWS). In such systems, ontologies are used to provide specification of semantics of web services, which is useful for their discovery and selection.

Ontologies are also used to annotate web services in Sheth (2003). Another issue that can be addressed when defining composite services is checking whether web services can semantically work together. In Corcho *et al.* (2003), the authors consider that semantic web services are modelled as problem-solving methods that describe web services and propose a framework to compose semantic web services. However, the semantic reconciliation between web services to compose has not received much attention yet. In Xu *et al.* (2004), the authors proposed a unified web services capability matching model. In this model, 'satisfaction' evaluation can be performed in a uniform way based on semantic interpretations of concepts. Recently, some attempts have been made towards using data mediation approaches to address web services mediation issues.

In Spencer and Liu (2004), the authors consider the problem of heterogeneous data models associated with inputs and outputs of web services to compose. They propose a rule-based approach to match semantically the outputs and inputs of web services. Multiple data transformation rules are defined using a description-logic reasoning system to analyse OWL-S and WSDL descriptions. Our approach has several points in common with this approach. We both focus on data heterogeneities between web services inputs and outputs. The main difference lies in the formalism used to provide active component to ensure semantic reconciliation between heterogeneous web services. In addition, we expect to improve our mediation mechanism to ensure semantic data exchange as is similarly done in semantic inter-operability of classical information system (Sciore *et al.*, 1994).

6.2 Adding context to web services

In the area of web services, context has been recently investigated in many research projects. The main objective of these projects is to facilitate the development and deployment of context-aware and adaptable web services. Standard web services descriptions are augmented with context information (*e.g.*, location, time, user profile, *etc.*) and new frameworks to support this are developed. The approach proposed in Kouadri-Mostéfaoui and Hirsbrunner (2003) is intended to provide an enhancement of WSDL language with context-aware features. The proposed Context-based Web Service Description Language (CWSDL) adds to WSDL a new part called Context Function. Context Function is used in order to select the best service offers when more than one service are discovered. This function represents the sensitivity of the service to context information. Another interesting approach was proposed in Keidl and Kemper (2004) to deal with context in web services. The approach consists of two parts: a context infrastructure and a context type set. The context infrastructure allows context information to be transmitted as a SOAP header-block within the SOAP messages. Each SOAP header-block represents one context information type. The context type set represents all context information that can be managed (*e.g.*, location, client context, *etc.*). The principal feature of this approach resides in the fact that the SOAP header-block is optional. This allows the use of legacy web services (*e.g.*, web services without context information) and in the fact that the context type set is extensible, which permits to deal with new context information.

However, the main difference between all these approaches and our proposed approach lies in the way semantics and context is treated. As far as we know, semantic and context have never been combined in an integrated approach to offer mediation and adaptability mechanisms for web services composition. Our proposed approach combines the ideas from both context-aware computing and data integration and attempts, as well, to adapt them to satisfy the needs/requirements of web services composition in a dynamic context.

7 Conclusion and perspectives

In this paper, we presented a semantic- and context-based approach that supports web services composition. We combined different emerging concepts such as mediation, ontologies and context to enable the semantic reconciliation among heterogeneous web services and to make composite web services context aware. Several development opportunities of composite web services come out after presenting our approach. First, context could be used to enable semantic data exchange between web services. This will add more semantics to web services descriptions. Second, we need to consider the possibility of generating mappings between ontologies in an automatic way. Third, it is necessary to implement the different components of the proposed framework in order to develop real applications based on composite web services.

While much of the work on the field of web services to date has focused on low-level standards for publishing, discovering and invoking web services, there are significant developments happening in this field. Indeed, it is worth mentioning conversation-driven composition of web services (Maamar *et al.*, 2004b), wireless web services (known also as M-services) (Maamar and Mansoor, 2002–2003), and model-driven approaches for web services composition (Baïna *et al.*, 2004). In this paper, we shed light on another major development, namely, semantic web services. These semantic web services are a cornerstone to the success of the semantic web strategy that aims at improving the technology used for organising, searching, integrating and allowing web-accessible resources to evolve (*e.g.*, documents, data). This requires the use of rich and machine -understandable abstractions for the representation of resource semantics.

References

Arroyo, S. and Stollberg, M. (2004) 'WSMO primer', *WSMO Deliverable D3.1, DERI Working DRAFT*, http://www.wsmo.org/2004/d3/d3.1/

Baïna, K., Benatallah, B., Casati, B. and Toumani, F. (2004) 'Model-driven web service development', *Proceedings of the 16th International Conference on Advanced Information Systems (CaiSE'2004)*, Riga, Latvia.

Benatallah, B., Sheng, Q.Z. and Dumas, M. (2003) 'The self-serv environment for web services composition', *IEEE Internet Computing*, January–February, Vol. 7, No. 1.

Benslimane, D., Roussey, C., Vangenot, C. and Arara, A. (2003) 'Towards a contextual content of ontologies', *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems (ISMIS'2003)*, Maebashi City, Japan.

Berardi, D., Calvanese, D., De Giacomo, G., Lenzerini, M. and Mecella, M. (2003) 'A foundational vision for e-Services', *Proceedings of the Workshop on Web Services, e-Business, and the Semantic Web (WES'2003) Held in Conjunction with the 15th Conference on Advanced Information Systems Engineering (CaiSE'2003)*, Klagenfurt/Velden, Austria.

Bouquet, P., Giunchiglia, F., Harmelen, F.V., Serafini, L. and Stuckenschmidt, H. (2004) 'Contextualizing ontologies', *Journal of Web Semantics*, October, Vol. 1, No. 4.

Corcho, Ó., Fernández-López, M., Gómez-Pérez, A. and Lama, M. (2003) 'ODE SWS: a semantic web service development environment', *Proceedings of the 1st International Workshop on Semantic Web and Databases Held in Conjunction with the 29th International Conference on Very Large Data Bases (VLDB'2003)*, Berlin, Germany.

Curbera, F., Khalaf, R., Mukhi, N., Tai, S. and Weerawarana, S. (2003) 'The next step in web services', *Communications of the ACM*, October, Vol. 46, No. 10.

Dean, M., Connolly, D., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. and Stein, L.A. (2002) *Web Ontology Language (OWL)*, http://www.w3.org/TR/2002/WD-owl-ref-20021112

Dey, A.K., Abowd, G.D. and Salber, D. (2001) 'A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications', *Human-Computer Interaction Journal, Special Issue on Context-Aware Computing*, Vol. 16, No. 1.

Gruber, T. (2000) *What is an Ontology?*, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

Keidl, M. and Kemper, A. (2004) 'A framework for context-aware adaptable web services', *Proceedings of the International Conference on Extending Database Technology (EDBT'2004)*, Crete, Greece.

Kouadri-Mostéfaoui, S. and Hirsbrunner, B. (2003) 'Towards a context-based service composition framework', *Proceedings of the 2003 International Conference on Web Services (ICWS'2003)*, Las Vegas, USA.

Leymann, F. (2001) 'Web services flow language', *Technical Report*, IBM Corporation.

Maamar, Z. and Mansoor, W. (2002–2003) 'Design and development of a software agent-based and mobile service-oriented environment', *e-Service Journal*, Indiana University Press, Vol. 2, No. 3.

Maamar, Z., Al-Khatib, G., Kouadri-Mostéfaoui, S., Lahkim, M. and Mansoor, W. (2004a) 'Context-based personalization of web services composition and provisioning', *Proceedings of the 30th IEEE EUROMICRO Conference – Software Process and Product Improvement*, Rennes, France.

Maamar, Z., Sheng, Q.Z. and Benatallah, B. (2004b) 'Towards a conversation-driven composition of web services', *Web Intelligence and Agent Systems: An International Journal*, Vol. 2, No. 2, ISSN 1570–1263.

Maamar, Z., Kouadri-Mostéfaoui, S. and Yahyaoui, H. (2005a) 'Towards an agent-based and context-oriented approach for web services composition', *IEEE Transactions on Knowledge and Data Engineering*, May, Vol. 17, No. 5.

Maamar, Z., Narendra, N.C. and van den Heuvel, W.J. (2005b) 'Towards an ontology-based approach for specifying contexts of web services', *Proceedings of the Montreal Conference on e-Technologies (MCETECH'2005)*, Montreal, Canada.

Madhavan, J., Bernstein Philip, A. and Rahm, E. (2001) 'Generic schema matching with Cupid', *Proceedings of the 27th Conference on Very Large Data Bases (VLDB'2001)*, Roma, Italy.

Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N. and Sycara, K. (2004) 'Bringing semantics to web services: the OWL-S approach', *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'2004)*, California, USA.

Medjahed, B., Rezgui, A., Bouguettaya, A. and Ouzzani, M. (2003) 'Infrastructure for e-Government web services', *IEEE Internet Computing*, January–February, Vol. 7, No. 1.

Mena, E., Illarramendi, A., Kashyap, V. and Sheth, A.P. (2000) 'OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies', *Distributed and Parallel Databases Journal*, Vol. 8, No. 2.

Milanovic, N. and Malek, M. (2004) 'Current solutions for web service composition', *IEEE Internet Computing*, November–December, Vol. 8, No. 6.

Mrissa, M., Ghedira, C., Benslimane, D., Maamar, Z. and Fayolle, J. (2005) 'A mediation framework for web services in a peer-to-peer environment', *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'2005)*, Cairo, Egypt.

Patil, A. and Oundhakar, S. (2002) *Semantic Annotation of Web Services (SAWS)*, Visited March 2005, http://www.arches.uga.edu/~abhijitp/SemEnt/SAWS-Final.htm

Patil A., Oundhakar, S., Sheth, A. and Verma, K. (2004) 'METEOR-S web service annotation framework', *Proceeding of the 13th International World Wide Web Conference (WWW'2004)*, New York, USA.

Power, R., Levis, D., O'Sullivan, D., Conclan, O. and Wade, V. (2004) 'A context information service using ontology-based queries', *Proceedings of the 1st International Workshop on Advanced Context Modelling, Reasoning, and Management Held in Conjunction with the 6th International Conference on Ubiquitous Computing (UbiComp'2004)*, Nottingham, UK.

Rios, D., Costa, P.D., Guizzardi, G., Ferreira Pires, L., Pereira Filho, J.G. and van Sinderen, M. (2003) 'Using ontologies for modeling context-aware service platforms', *Proceedings of the Workshop on Ontologies to Complement Software Architectures Held in Conjunction with the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'2003)*, California, USA.

Satyanarayanan, M. (2001) 'Pervasive computing: vision and challenges', *IEEE Personal Communications*, August, Vol. 8, No. 4.

Schilit, B., Adams, N. and Want, R. (1994) 'Context-aware computing applications', *Proceedings of the 1st IEEE Workshop on Mobile Computing Systems and Applications*, California, USA.

Sciore, E., Siegel, M. and Rosenthal, A. (1994) 'Using semantic values to facilitate interoperability among heterogeneous information systems', *ACM Transactions on Database Systems*, June, Vol. 19, No. 2.

Sheth, A.P. and Ramakrishnan, C. (2003) 'Semantic (web) technology in action: ontology driven information systems for search, integration and analysis', *IEEE Data Engineering Bulletin*, Vol. 26, No. 4.

Spencer, B. and Liu, S. (2004) 'Inferring data transformation rules to integrate semantic web services', *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan: Springer Verlag, 7–11 November, LNCS 3298.

Wiederhold, G. (1992) 'Mediators in the architecture of future information systems', *IEEE Computer*, March, Vol. 25, No. 3.

Xu, C., Cheung, S.C. and Xiao, X. (2004) 'Semantic interpretation and matching of web services', *Proceedings of the 23rd International Conference on Conceptual Modeling (ER'2004)*, Shanghai, China.