# AN ONTOLOGY-BASED SERVICES COMPOSITION ALGORITHM

Jean-Charles Verdié
*Sand Labs, Bt. 6, Parc Club du Millénaire, 1025 Rue H. Becquerel, 34000 Montpellier*

Danièle Hérin, Michel Sala
*Université Montpellier II, LIRMM, 161 Rue Ada, 34392 Montpellier Cedex 5*

Keywords: Web Services, Composition.

Abstract: In the Semantic Web Services area, completing a complex request means calling many Web Services. They are characterized by their heterogeneity since they are build independently from the context in which they will be used. To compose them, we need to consider annotation and meta-data which will allow their characterization. The goal of this paper is to propose the execution of a formal model, allowing the automated processing of the composition of many Semantic Web Services. We define a request as a sequence of different Web Services with the help of a formal logic. We define the paths allowing its concrete resolution. Thanks to an abstraction of the Services whether they are viewed as physical, local or global, we propose an automation of the building of the request resolution map.

## 1 INTRODUCTION

A Web service is a software component that takes the input data and produces the output data (Rao, 2004). It usually comes with an interface understandable by a machine, specifically WSDL (W3C, WSDL, 2004) or an human (HTML forms).

In this paper, we focus on services without prejudice of the way they are called. It could be via a SOAP envelope (W3C, SOAP, 2003) providing the information requested by the Service WSDL interface, or through a human-machine interface with forms. We formalize a Web Service as a transaction, related to an ontology of concepts. This Web Service describes how to compare, classify, process the data of this ontology. By Transaction, we mean a software process of data exchange and processing on the domain ontology E, which associates a subset $E_d$ of E to a subset $E_f$.

Processing a request formulated by the user often involves the composition of many Web Services, and the consideration of constraints between these Services. These Web Services and these constraints are described in a generic way with a global ontology, which needs to be translated to a specific ontology to fit the description of the Service on a specific Web site, and eventually translate to a physical level, with the scenario of page sequencing, or the creation of a SOAP envelope.

We propose a model to formalize this approach, and generate a resolution map taking into account the Services operated and the constraints expressed.

All along this paper, we will illustrate our methodology with the following request:

***$R_1$*: Book an hotel room and a plane seat to go to the city where the hotel is.**

*Completing $R_1$ means composing two services:*

- $WS_1$: book an hotel room
- $WS_2$: book a plane seat
- Constraint between the two transactions: the plane must land in the destination city the day when the hotel booking starts.

The remainder of the paper is as follow:

- In the chapter 2 we present a formal approach of composition, with three abstraction levels.
- Chapter 3 describes the detection of services and their constraints.
- The resolution map generation is presented in chapter 4.

Verdié J., Hérin D. and Sala M. (2006).
AN ONTOLOGY-BASED SERVICES COMPOSITION ALGORITHM.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 461-464
DOI: 10.5220/0001240004610464

# 2 CONCEPTUAL MODELLING OF A COMPOSITION

According to (Payne, 2004) the Semantic Web technologies use knowledge representation methods in a distributed environment. Semantic Web Services, as the new research paradigm, are usually defined as a improvement of Web Services with semantic annotations. But these services are based on different ontologies. The requester agent has to translate the description of every service from its own ontology into the Web Service's one to produce a valid requests.

Our approach proposes to solve this problematic. We propose a model in three levels: global, local and physical.

The global level sees a Service as an application from a set of data E in itself.

At the local level, a Service is an application which associates a data set equivalent to $E_D$, to a data set equivalent to $E_R$.

At the physical level, a Service consists in the sequence of pages on which we submit or retrieve data semantically equivalent to those in the union of $E_D$ and $E_R$.

We will now present these three levels with more details, then we will explain how we can translate data or transactions from a level to another.

## 2.1 Global Level

Let E be the set of concepts expressed when processing a given Web Service Abstracted, WSA. We build an ontology for this service, that we name *global data model* (GO), and which could be described in OWL. We name it Abstracted since it is not related to a concrete implementation of the Service.

We use a simplified ontology based on two relationships: generalization and aggregation.

## 2.2 Local Level

On every web site registered as addressing a service WSA, we have a *local ontology (LO)*. It is a specialization of the OWL model described at the global level.

A site processes a service if there is a morphism between the global service WSA and (GO) and a Local Web Service WSL and (LO).

## 2.3 Physical Level

Concretely, a Web site without SOAP/WSDL automation completes a Service by browsing a sequence of pages. On each of these pages, forms (the most often) allow to express some concepts, the actual subset of the local ontology.

We browse a sequence pages so that the aggregation of the concepts expressed is equal to the input data set of the transaction, then another sequence whose concepts aggregation contains the output data subset. We name this sequence a path.

## 2.4 Formalizing the Three Abstraction Levels

We associate a table such as Table 1 to every WSA. We identify the ontology of the service, and the transactions needed to complete this service. This global level is completed by local and physical vision of each service.

For instance, for Service $WS_2$, *booking a plane seat*, each flight company may, from the concepts of "departure city", "arrival city" and "departure date" return the information of availability.

Depending on the concrete architecture of the flight company site, the path allowing the realisation of this service won't be the same. The table below expresses two different sites performing the same service, one through a simple request, the other through a composite request.

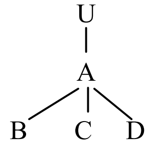
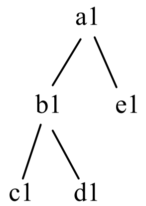
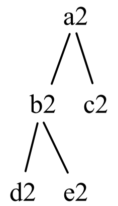
# 3 DETECTION OF WEB SERVICES, CONSTRAINTS, PHYSICAL COMPOSITION

Constraints between Services are the translation, from an event standpoint, of user constraints, i.e. the translation of a constraint in precedence, validation and triggering between different services. These constraints are formalized as guards (Singh, 1996).

We propose a methodology to build the physical resolition map, **Algorithm 1:**

- **Step 1**: Identify services to operate to complete the composition
- **Step 2:** Identify constraints between these services
- **Step 3:** Formalize the resolution of each Service

Table 1: Three-levels modelization of an ontology.

| Level | Data | | Processing | |
|---|---|---|---|---|
| GLOBAL<br><br>Global Ontology (GO) |  | | $WSA_1$ A, B → C, D | |
| | Site 1 | Site 2 | Site 1 | Site 2 |
| LOCAL<br><br>Local Ontology (LO) |  |  | $WSL_1^1$ a1 → b1<br><br>$WSL_2^1$ a1,b1→ a1, b1 | $WSL_2^2$ a2, b2 → c2, d2 |
| PHYSICAL | [a1] $p_1^1$<br>[a1 b1] $p_2^1$<br>[a1 b1 c1 d1] $p_3^1$ | [a2 b2] $p_1^2$<br>[a2 b2 c2 d2] $p_2^2$ | $c_1^1$ =<br>$p_1^1 \rightarrow p_2^1 \rightarrow p_3^1$ | $c_1^2$ =<br>$p_1^2 \rightarrow p_2^2$ |

**Step 4:** Identify web sites that are likely to address a service: *Methods such as UDDI (UDDI, 2005) allow that kind of identification. Another method consists in using a broker based upon the ServiceProfile of the Web Service described in OWL-S (Paolucci, 2005), or even on using a confidence level assessment shared by peer-to-peer.*

- **Step 5**: For each site S, generate the path C allowing the completion of the service *(see next chapter)*.
- **Step 6**: Calculate, using all paths and constraints, the resolution map of the composition *(see next chapter)*.

## 4 CALCULATION OF THE RESOLUTION MAP (PHYSICAL LEVEL)

For each Local Web Service WSL, we associate a path representing a sequence of pages *C* on a site addressing this service.

*For instance, let $E_D$ and $E_R$ be two set of data:*
*$WSL_1$: $E_D \rightarrow E_R$*
*$C_1$: $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$*

*The path $C_1$ enables the service $WSL_1$ on a site $S_1$. It means that there are $c_D$ and $c_F$, sub-path of C so that there is an equivalence between $\cup c_D$ and $E_D$, and between $\cup c_F$ and $E_R$.*

The resolution map is build using A**lgorithm 2**:

- For each page $p_k$ in $\cup c_i$ , union of all paths, do:

- Place all pages $p_k$, and according to the links expressed in $c_i$ (i.e. sequences), place links.
- For each $d_j$ in the set of constraints between two transactions converted to pages, $d_j = \{p_1 \rightarrow p_2\}$, place the expressed link, equivalent to a transactional guard.
- If $p_k$ owns n predecessors $p_{k1}$ and $p_{k2,}$ then place a transition between the n predecessors and $p_k$.

## 5 EXISTING APPROACHES AND OUR POSITIONING

We position our approach with respect to works completed in the area of relaxed data and transactions, process management and workflows. Many process management methods have been proposed, including Petri networks, Allen algebra, workflows and Component-Based Software Programming. Rao (Rao, 2004) defines a **Dynamic Workflow** model based upon the creation of a process model followed by constraints specification.

## 6 CONCLUSION

We formalized a model allowing to consider a morphism between Services defined at a global level and physical sequence of web pages. This approach is particularly suitable to services without machine-machine interface and we are currently prototyping it.

Our future works direction are:

- Adapting this formal model to Semantic Web technologies (OWL) and Web Services (SOAP, WSDL).
- Formalize projections between the three abstraction levels Global, Local and Physical.
- Search and identify agents delivering services, using meta-data and annotations through ontology.

## REFERENCES

German, D.M., 2003. Introduction to Transaction Management, slides C SC 370 Apptus. In apptus.se/upload/PDF/ACID_Transaction_Management.pdf

Paolucci, M.,Soudry, J., Srinivasan, N., Sycara, K., 2005. A broker for OWL-S Web Services. In *The Robotics Institute,* Carnegie Mellon University.

Rao, J., Su, X., 2004. A Survey of Automated Web Service Composition Methods. In *Proceedings of SWSWPC 2004,* San Diego, California, USA.

Singh, M., 1996. Formal Semantics of Workflow Computations. In *TR-96-08,* NCSU CS.

Singh, M., Huhns, M.N., 2005. Service-Oriented Computing: Semantics, Processes, Agents. In *J. Wiley & Sons, LTD*

UDDI, 2005. White Paper. In *uddi.org/whitepapers.html*

W3C, SOAP, 2003. Version 1.2 Part 0: Primer, Recommendation. In *www.w3.org/TR/soap12-part0/*

W3C, WSDL, 2004. Web Services Description Langage. In www.w3.org/TR/wsdl