Proceedings of I-KNOW '09 and I-SEMANTICS '09
2-4 September 2009, Graz, Austria

A Proposed Ontology-based Algebra For Service Annotation, Discovery, and Composition

Nadia Yacoubi and Yann Pollet
(Conservatoire National des Arts et Métiers
{nadia.yacoubi, yann.pollet}@cnam.fr)

Mohamed Ben ahmed
(Laboratoire RIADI, National School of Computer Science
mohamed.benahmed@riadi.rnu.tn)

Abstract: Service discovery and service composition are two crucial issues in the emerging area of Service-oriented Computing (SoC). The introduction of semantic annotations in Web service descriptions can support the discovery process of services and guide their composition into workflows. In this paper, we propose a *Structured Relationship Algebra* that extends existing frameworks for semantic service descriptions by a set of algebraic operators and allows to define in a uniform way formal semantic annotations regardless the framework adopted to describe service semantics. The proposed algebra is comprised by a set of three algebraic operators and allows to define new objet properties at the ontology level called *derived properties*. Derived properties are used to describe the data transformation processed by a service by associating constraints on both properties and instances of the concepts and their related ontological relationships.
Key Words: Semantic Web services, ontology-based algebra, formal semantic annotations, service discovery, service composition.
Categories: H.3.5., I.2.4

1 Introduction

Service-oriented Computing (SoC) is emerging as a new, promising computing paradigm that centers on the notion of service as the fundamental element for accessing heterogeneous, rich and distributed resources in an interoperable way [PTDL07]. Web services are self-describing components that support a rapid and significant reuse of distributed applications. They are offered by service providers, which procure service implementation and maintenance, and supply service descriptions. On the other side, semantic annotations of Web services have several applications in the construction and management of service-oriented applications [MSZ01]. As well as assisting in the discovery of services [BHL+05], such annotations can be used to support the user in composing services into workflows [CS03]. Thus, several frameworks for semantic Web services has been proposed such as the OWL-S Semantic Markup for Web services [MBM+07], the Web Service Modeling Ontology (WSMO) [RKL+05], and the Semantic Annotations for WSDL (SAWSDL) [KVBF07]. These frameworks aim to create semantic annotations to describe *functional* and *non-functional* properties of the service. In this paper, we present an algebra that extends existing frameworks for semantic service descriptions by a set of algebraic operators allowing to define in a uniform way formal annotations regardless the semantic framework adopted to describe service semantics. We present the *Structured Relationship Algebra* that is comprised by a set of three algebraic operators allowing to define new semantic relationships at the ontology level called *derived properties*. Using algebraic operators, *derived properties* are created by associating constraints on both properties and instances of the concepts participating to the relationships. We show how *derived properties* can be used to describe the relations that relate the inputs of a service to its outputs.

The remainder of this paper is organized as follows. Section 2 presents the preliminaries of the proposed approach. In Section 3, we describe the three algebraic operators and define their properties and their use to define semantic annotations. Finally, section 4 concludes the paper.

2 Preliminaries

In this section, we present some preliminaries of the proposed algebra. First, we introduce the notion of native ontology. A native ontology is a domain ontology described using an ontology definition language such as OWL[1], RDF[2], or RDF-S[3]. To simplify, we model ontologies as graphs of concepts linked by a set of directed relationships. The initial ontology graph is called a Native Ontology Graph (NOG), respectively defined with native concepts and relationships. The graph extended with derived properties is called Derived Ontology Graph (DOG).

Definition 1 Native Ontology Graph. A Native Ontology Graph is a directed labeled graph O=(T_C,T_R) where T_C is a finite set of labeled classes (nodes) and T_R is a finite set of labelled relationships (edges). A relationship r is written as (r, c_1, c_2) where dom(r)=c_1, range(r)=c_2, c_1 and $c_2 \in T_C$, and r $\in T_R$.

Example 1. The figure 2 illustrates a fragment of a domain ontology. In a given regional area, we have different healthcare institutions or practitioner offices. Each institutions have his own medical files of their patients, stored as medical records. Each medical practitioner in the institution has a given role in the healthcare process of a patient. The role inside a given patient healthcare process specify the access rights to the patient information.

[1] http://www.w3.org/TR/owl-ref/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/rdf-schema/

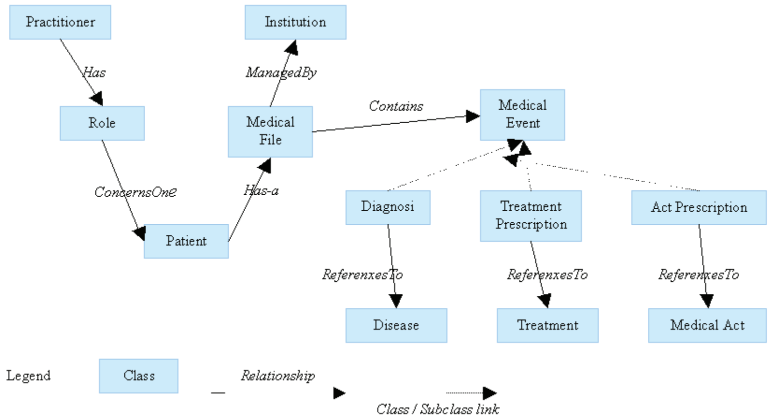

Figure 1: A Health Care Domain ontology

3 Our approach

The algebra defines two binary operators and one unary operator applied on relationships: composition, union, and restriction.

3.1 Composition operator

The composition operator is a binary operator that allows to create a derived object property given two object properties. The operator is defined as follows and illustrated in Figure 2.

Definition 2 Composition Operator. Given two relationships R_1 and R_2 in T_R, such as dom(R_1)=C_{from1}, range(R_1)=C_{to1}, dom(R_2)=C_{from2}, range(R_2)= C_{to2}, the composition R = $R_1 \otimes R_2$ is defined by $\langle$R, x, z$\rangle$ iff it exists an individual c belonging to both C_{to1} and C_{from2} such as $\langle R_1$, x, c$\rangle$ AND $\langle R_2$, c, z$\rangle$, x, c, and z are respectively instances of C_{from1}, C_{to1} MIN C_{from2}, and C_{to2}.

If C_{to1}=C_{from2} then we get the classical operator of composition. If C_{to1} MIN $C_{from2} = \emptyset$ then R is the *null* relationship. The composition operator is not *commutative*. It is *associative* and it exists an identity element that is the *identity* relationship defined on the domain class *Universal*=$\cup_i$ C_i, C_i in T_C that corresponds to each individual itself such as $\langle Identity, o, o\rangle$ for any individual o.

Example 2. Figure 2 depicts a fragment of a domain health care ontology. The fragment is extended with derived object properties that are the result of composing some relationships.

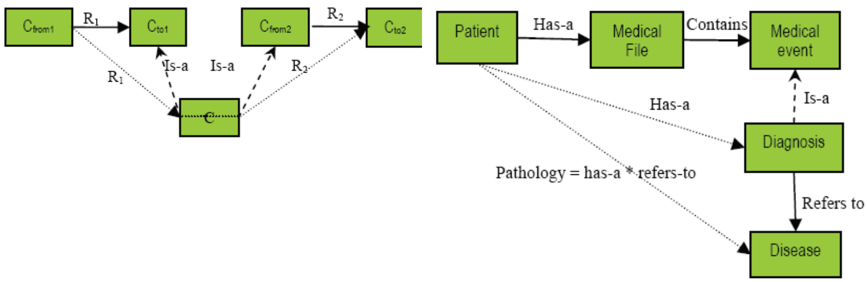

Figure 2: Illustration of the composition operator

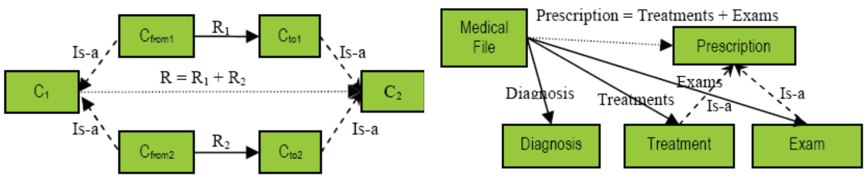

Figure 3: Illustration of the union operator

3.2 Union operator

The second operator of the algebra is the union operator denoted by $\oplus$, the union operator creates a derived object property by the union of two object properties as follows.

Definition 3 Union Operator. Given two relationships R_1 and R_2 such as such as dom(R_1)=C_{from1}, range(R_1)=C_{to1}, dom(R_2)=C_{from2}, range(R_2)=C_{to2}, the union R = $R_1 \oplus R_2$ is defined by $\langle$R,x,y$\rangle$ iff $\langle R_1$,x,y$\rangle$ OR $\langle R_2$,x,y$\rangle$ such as dom($R_1 \oplus R_2$)= C_{from1} MAX C_{from2}, and range($R_1 \oplus R_2$)= C_{to1} MAX C_{to2}.

The union operator is *commutative* and *associative*. Note that the composition operator is *distributive* related to the Union operator. Finally, the *idempotence* property is that union of the same relationship is itself, i.e., R = R $\oplus$ R.

Example 3. Figure 3 illustrates the use of the union operator defining a new object property on the medical file concept, standing for all medical events corresponding to medical prescription that are either Treatment either exams.

3.3 Algebraic expression with composition and union operators

The composition and union operators defined above enable to write algebraic expressions, using a set of operands R_1, R_2, ...,R_n that are native and/or defined derived relationships. There is no restriction on possible composition and union, as these operations are always defined for any ordered pair of operand relationships. A composition or union operation with no meaning delivers an $EMPTY$ relationship.

If we consider an algebraic expression R = Expr (R_1,...,R_n) where $R_{i=1,...,n}$ are either native and/or derived relationships, this expression defines a new relationship R, with calculable domain and range thanks to the formal semantics of the union and composition operators. The expression becomes an expression of native relationships only, if we substitute each operand derived relationship by its definition in function of native relationships (iteratively in a finite number of times, as the number of definition is finite and as no recursion is possible in definitions).

This expression defines a multipath in the native ontology graph starting from an origin class C_{from} and ending at a destination class C_{to}. For a given individual o, it will be possible to calculate all the correspondents of o by R (i.e., all the individual $o^{'}$ such as $\langle$R, o, $o^{'}\rangle$) if we are able to determine all the correspondences for any relationship R_i in the expression of R. However, in practice, we may also need to follow a path in the graph starting from a concept C, but with an interest for a subset of instances of a class $C^{'}$ only. It is not possible to represent such a *restriction* with simple multipaths made of sequences (composition) and parallel (union) edges. To represent such restrictions, we define constraints on the properties instances of the corresponding range and domain concepts of a relationship.

3.4 Restriction operator

In addition to the composition and union operators, we define the unary restriction operator that associates to any relationship R a derived relationship $R^{'}$ that has as a domain a sub-domain of R, and limiting the possible correspondence to a sub-range of R. Sub-domain and sub-range are defined by two restriction predicates, called PRE and POST conditions. This predicates being fixed, the restriction operator is considered as a unary operator applying on relationships.

Definition 4. Given two classes C_1 and C_2, a relationship R such as dom (R) = C_1 and range (R) = C_2, two predicates PRE and POST respectively defined on the properties of C and C, the restriction operator defined by the 3-tuple (R, PRE, POST) creates a new relationship $R^{'}$= RESTRICT (R, PRE, POST) such as $\langle R^{'}$, o, $o\rangle$ iff $\langle$R, o, $o^{'}\rangle$ AND PRE(o) is true AND POST($o^{'}$) is true.

The restricted relationship $R^{'}$ associates to an individual o from C its correspondent $o^{'}$ given by the relationship R iff the individual o verify the predicate PRE and the individual $o^{'}$ verify the predicate POST. If o does not belong to C^{PRE} (the set of instances of the class C that verify PRE), then it has no correspondent and the restricted relationship $R^{'}$ is *empty*. If o belongs to C^{PRE}, but $o^{'}$ does not belong to C^{POST} (the set of instances of the class that verify POST), then o has no correspondent by the relationship $R^{'}$. The result $R^{'}$ = RESTRICT (R, PRE, POST) will be denoted $R^{PRE,POST}$ such as dom($R^{PRE,POST}$) = C^{PRE} and range($R^{PRE,POST}$) = C^{POST}. The figure 7 below illustrates the restriction operator.

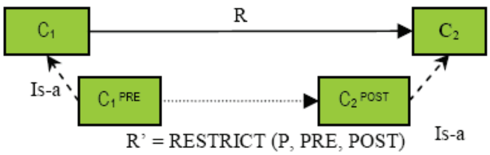

Figure 4: Illustration of the restriction operator

Definition 5 Semantic Mapping Assertion. Given a domain ontology O= (T_C,T_R) and a set of services S_i, For each S_i, a semantic assertion mapping is defined as 3-tuple ⟨MAP, S_i, $R^{PRE,POST}$⟩ such as R is an algebraic expression over T_R constructed using the union, composition operators and PRE and POST are two logical predicates applied respectively on dom(R) and range(R) using the restriction operator.

Example 4. Let DS_1 a Web service that delivers information about patients and their diagnosis information disease. The algebraic expression that express the service semantics is interpreted as follows:

E(DS_1) = has(Patient, Medical_File) ⊗ *contains(Medical_File, Diagnosis)* ⊗ *refers(Diagnosis, Disease).*

Let DS_2 a Web service that delivers therapy prescription patients including medical acts. The algebraic expression that express the service semantics is interpreted as follows:

E(DS_2) = has(Patient, Medical_File) ⊗ *[contains(Medical_File, Therapy _Prescription)* ⊕ *contains(Medical_File, Act_Prescription)].*

4 Conclusions and Future Work

In this paper, we proposed the SRA algebra comprised by a set of algebraic operators. The SRA operators allow to define formal semantic annotations that capture the data transformation process performed by services. The composition and union operators allow to declare multipaths through a domain ontology graph as a sequence or the union of several semantic relationships. The restriction operator allows to declare a new relationship by defining restrictions on the domain and range of an input one. As future work, we plan to define algorithms that deduces service composition plans given a query defined as an algebraic expression. Also, we plan to extend the proposed algebra with reasoning capabilities in order to enhance the composition and discovery process.

References

[BHL+05] Benatallah, B., Hacid, M.-S., Leger, A., Rey, C., and Toumani, F.: On automating web services discovery; *The VLDB Journal*, 14(1):84–96, 2005.

[CS03] Cardoso, J. and Sheth, A.: Semantic e-workflow composition; *J. Intell. Inf. Syst.*, 21(3):191–225, 2003.

[KVBF07] Kopecký, J., Vitvar, T., Bournez, C., and Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema; *IEEE Internet Computing*, 11(6):60–67, 2007.

[MBM+07] Martin, D., Burstein, M., Mcdermott, D., Mcilraith, S., Paolucci, M., Sycara, K., Mcguinness, D. L., Sirin, E., and Srinivasan, N.: Bringing Semantics to Web Services with OWL-S; *World Wide Web*, 10(3):243–277, 2007.

[MSZ01] McIlraith, S. A., Son, T. C., and Zeng, H.: Semantic web services; *IEEE intelligent systems*, 16(2):46–53, 2001.

[PTDL07] Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F.: Service-oriented computing: State of the art and research challenges; *Computer*, 40(11):38–45, 2007.

[RKL+05] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D.: Web Service Modeling Ontology; *Applied Ontology*, 1(1):77–106, 2005.