Path-Based Semantic Annotation for Web Service Discovery

Julius Köpke, Dominik Joham, and Johann Eder

Department of Informatics-Systems, Alpen-Adria-Universität Klagenfurt, Austria
{julius.koepke|johann.eder}@aau.at, dominik.joham@gmail.com
http://isys.uni-klu.ac.at

Abstract. Annotation paths are a new method for semantic annotation which overcomes the limited expressiveness of semantic annotations by concept references as defined in the SAWSDL standard. In this work we show some preliminary evaluation of the feasibility of annotation paths for web service discovery. The experiments suggest that annotation paths can capture the semantics of XML schemas and web service descriptions more precisely and appears as a promising approach for improving web service discovery.

Keywords: Web Service Matching, Service Discovery, Semantic Annotation, SAWSDL, Annotation Paths, XML-Schema Matching, Semantic Matching

1 Introduction

Web service discovery aims at (semi-)automating the search for suitable web services. A web service discovery systems accepts a service request (a specification of the needed web service) and a set of web service descriptions (advertisements) as input and returns a list of web service descriptions ranked by relevance for the request. There are many different approaches ranging from the structural or lexical comparison of requests and advertisements to approaches that are based on the explicit definition of the semantics using ontologies [12]. We specifically address the usage of external knowledge provided by semantic annotations with a reference ontology using SAWSDL [6] annotations. The W3C recommendation SWASDL (Semantic Annotations for WSDL and XML-Schema) specifies a light-weight approach for the annotation of web services with arbitrary semantic models (e.g. ontologies). SAWSDL introduces additional attributes for XML-Schema and WSDL-documents. *ModelReferences* refer to ontology concepts and *Lifting-* and *Lowering-Mappings* refer to arbitrary scripts that transform the inputs and output XML-data to and from instances of some semantic model. $ModelReferences$ are proposed for service discovery, while *Lifting-* and *Lowering-Mappings* are proposed for service invocation and only apply to the annotation of inputs and outputs defined by XML-Schema.

Our previous work focused on the annotation of XML-schemas with reference ontologies in order to automate the generation of executable schema mappings

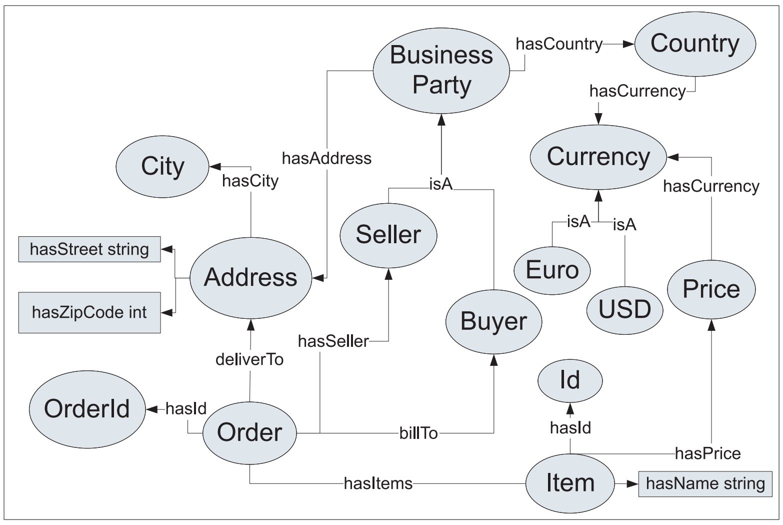


Fig. 1. Example reference ontology [8]

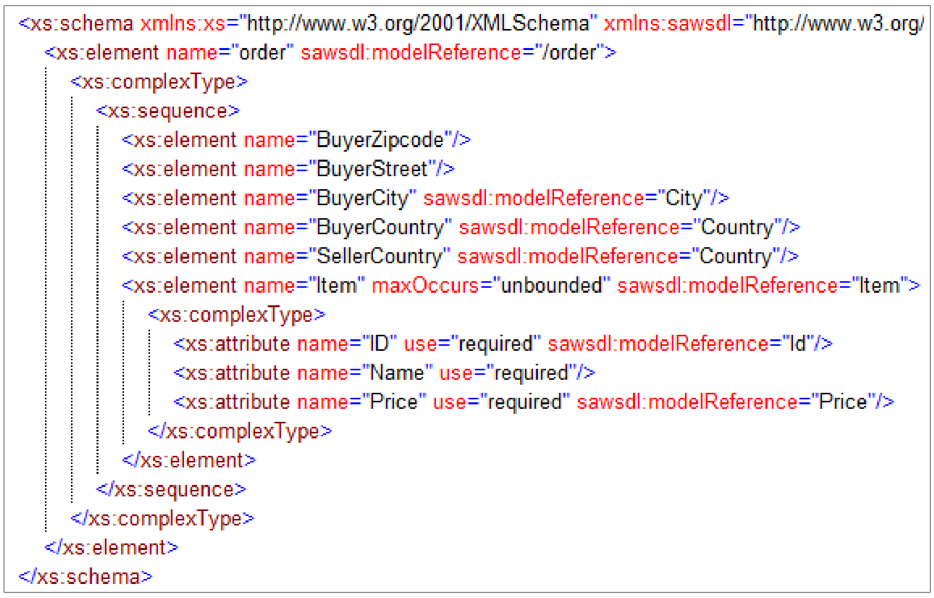
for document transformations [8–10, 7]. We could show that the expressiveness of SAWSDL is not sufficient for the generation of schema mappings when general reference ontologies are directly used for the annotation. Therefore, we have proposed an extended annotation method that is based on annotation paths rather than single concept annotations. Since this method already showed its usefulness for XML-document transformations [7] we assume that annotation paths can also improve web service discovery. The general hypothesis is that if the annotation method allows a more precise definition of the semantics then the precision of service matching for service discovery can be improved. Existing approaches for SAWSDL based service discovery such as [5, 3] can partly solve the problem of non precise semantic annotations by using additional dimensions such as structure or textual similarity.
To give a first answer on this hypothesis we discusses the usage of annotation paths for web service discovery and report some preliminary results.

2 Annotation Path Method

In some examples we show limitations of simple references to concepts for the annotation of arbitrary XML-schemas or web service descriptions with existing reference ontologies. We then present the general concept of annotation paths (for details we refer to [8]).

2.1 Example

The SAWSDL [6] standard addresses semantic annotations for both web service descriptions and for XML-schemas which are related since WSDL description



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/
  <xs:element name="order" sawsdl:modelReference="/order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BuyerZipcode"/>
        <xs:element name="BuyerStreet"/>
        <xs:element name="BuyerCity" sawsdl:modelReference="City"/>
        <xs:element name="BuyerCountry" sawsdl:modelReference="Country"/>
        <xs:element name="SellerCountry" sawsdl:modelReference="Country"/>
        <xs:element name="Item" maxOccurs="unbounded" sawsdl:modelReference="Item">
          <xs:complexType>
            <xs:attribute name="ID" use="required" sawsdl:modelReference="Id"/>
            <xs:attribute name="Name" use="required"/>
            <xs:attribute name="Price" use="required" sawsdl:modelReference="Price"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

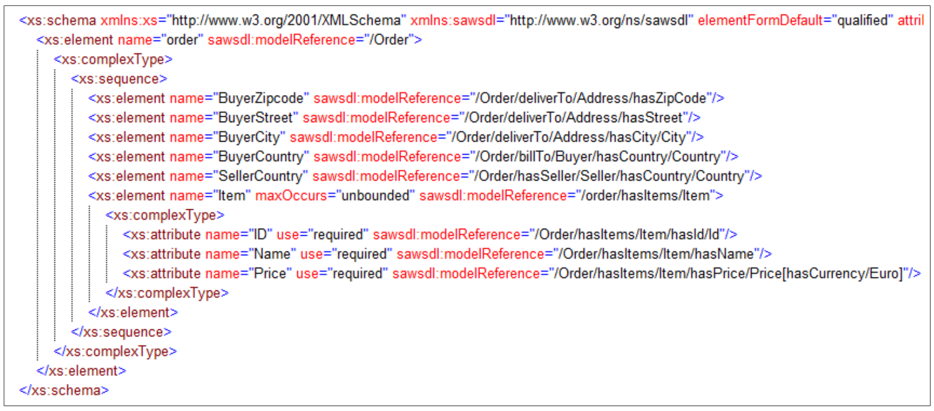
Fig. 2. Sample XML-Schema with model-references [8]

use XML-Schema to define the inputs and outputs of operations. The XML-Schema document shown in Figure 2 is annotated using simple concept references referring to the ontology shown in Figure 1 using the $sawsdl : ModelReference$ attribute.

The annotated document in Figure 1 exhibits the following problems:

- The elements *BuyerZipcode* and *BuyerStreet* cannot be annotated because the *zip-code* is modeled in form of a data-type property and not by a concept in the ontology.
- The *BuyerCountry* element is annotated with the concept *country*. This does not fully express the semantics because we do not know that the element should contain the country of the buying-party. In addition the *Seller-Country* element has exactly the same annotation and can therefore not be distinguished.
- The attribute *Price* is annotated with the concept *Price*. Unfortunately this does not capture the semantics. We do not know the subject of the price (an item) and we do not know the currency.

We have always used exactly one concept for the annotation in the example. However, SAWSDL supports lists of concepts in the $modelReference$ attribute but it does not allow to specify the relations between the concepts in this list. Therefore, this does not help to solve the shown problems. In the examples above we have only annotated data-carrying elements. If we would in addition also annotate the parent elements in this case the *order* element we could add a bit more semantic information. It would be clear that the annotations of the



```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/ns/sawsdl" elementFormDefault="qualified" attril
  <xs:element name="order" sawsdl:modelReference="/Order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BuyerZipcode" sawsdl:modelReference="/Order/deliverTo/Address/hasZipCode"/>
        <xs:element name="BuyerStreet" sawsdl:modelReference="/Order/deliverTo/Address/hasStreet"/>
        <xs:element name="BuyerCity" sawsdl:modelReference="/Order/deliverTo/Address/hasCity/City"/>
        <xs:element name="BuyerCountry" sawsdl:modelReference="/Order/billTo/Buyer/hasCountry/Country"/>
        <xs:element name="SellerCountry" sawsdl:modelReference="/Order/hasSeller/Seller/hasCountry/Country"/>
        <xs:element name="Item" maxOccurs="unbounded" sawsdl:modelReference="/order/hasItems/Item">
          <xs:complexType>
            <xs:attribute name="ID" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasId/Id"/>
            <xs:attribute name="Name" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasName"/>
            <xs:attribute name="Price" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasPrice/Price[hasCurrency/Euro]"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fig. 3. Sample XML-Schema document with Annotation-Path Method [8]

child-elements of the order-element can be seen in the context of an order. Unfortunately this would not help for the ambiguities between the *BuyerCountry*- and the *SellerCountry* element. In general it would require a strong structural relatedness between the ontology and the annotated XML-Schema or service description which we cannot guarantee when many different schemas or services are annotated with a single reference ontology. In addition SAWSDL does not define that there are any relations between the annotations of parent and child elements. A solution for these non precise annotations is the usage of a more specific reference ontology, which contains concepts that fully match the semantics of each annotated element. For example it would need to contain the concept *InvoiceBuyerCountry* and *InvoiceBuyerZipCode*. However, enhancing a general reference ontology with all possible combinations of concepts leads to a combinatorial explosion.

2.2 Annotation Path Method

We propose a new annotation method based on annotation path expressions that are sequences of steps referring to concepts and properties of a reference ontology. The first step of an annotation path is always a concept. The last step of an annotation path can be a concept or a data-type property. Between two concept steps there is always an object property step. Concept steps can have constraints denoted in square brackets. In Fig. 3 we give some examples for annotation paths - more details can be found in [8].

Annotation paths can automatically be represented in form of OWL2 concepts which can be used to extend the reference ontology. For example the path *Order/billTo/Buyer[Mr_Smith]/hasCountry/Country* is represented as a subclass of *Country* that has an inverse *hasCountry* relation to a *Buyer* whose name is *Mr. Smith* who has an inverse *billTo* relation to an *Order*. This can be represented by the OWL expression *Country and inv (hasCountry) some (Buyer and {Mr_Smith} and inv (billTo) some (Order))*.

The extraction of annotations from a schema requires to rewrite the schema in order to cope with reused elements first. The resulting schema may contain additional annotations. Since schema elements can refer to other schema elements and types, the full annotation path has to be concatenated from the annotation paths of the elements. For an example an XML-element *DeliveryAddress* is itself annotated with the annotation path */**Order**/deliverTo/**Address***. It has a type definition *address*. The address type itself contains various elements. One of them is *street* which is annotated with */**Address**/hasStreet*. In order to construct the complete semantics of the *street* element which is a child element of *DeliveryAddress* we get the additional path */**Order**/deliverTo/**Address**/hasStreet*.

3 Path-Based Service Matching Prototype

To apply the annotation path method to web service discovery we implemented a logics based service matcher [1] that operates only on path-based annotations of the inputs and outputs of operations. No other dimensions of the service descriptions are used for matching. The assumption is that two operations with the same inputs and outputs are likely to be the same operation. We do not address the annotation of operations themselves. In order to rank different web services according to a request we automatically generate one XML-Schema for the inputs and one XML-Schema for the outputs of each operation of the advertisements and the request. These schemas are then matched and an overall confidence value for the service match in the interval [0..1] is computed. The ranking is then based on the confidence values. The matching process of the schemas operates in 4 phases:

- *Annotation Path Extraction:* The input and output schemas of each operation are transformed to an internal tree representation where no types are reused using the COMA3[11] library. The annotation paths are rewritten as described in the last paragraph of Sect. 2.2. Finally all annotation paths are extracted from the resulting tree.
- *Extended Ontology Generation:* The annotations are transformed to OWL concepts and an extended reference ontology is created.
- *Matching and Mapping:* The XML-Schemas of the request and of each advertisement are matched based on the annotations using a standard OWL reasoner (pellet). Two schema elements $s1$ from the source schema and $t1$ from the target schema match if the annotation concept (the corresponding annotation path represented as an OWL concept) of $s1$ is equivalent to the annotation concept of $t1$ or if there is a subclass or superclass relation between $s1$ and $t1$. In case of equivalence the confidence value of the match is 1. In case of the subclass match the confidence value of the match is 0.8 weighted by the concept distance between the annotation concept of $s1$ and $t1$ in the extended reference ontology. In case of a superclass to subclass match the confidence value is 0.6 also weighted by the distance in the ontology. After the confidence values are computed for each combination of

elements of the source and target schema, a schema mapping is created based on the best matching elements.

– *Ranking:* Finally, an overall confidence value of each schema mapping is computed by aggregating the confidence values of the mapping elements using min, max or avg. strategies and the advertisements are ordered descending by the overall confidence values.

4 Evaluation

The goal of the evaluation is to provide preliminary results whether the annotation path method leads to better results in service discovery. Therefore, we have evaluated [1] our service matcher that exploits only path based semantic annotations against existing SAWSDL-based service matchers. The assumption is that when this simple service matcher can compete with state of the art service matchers that exploit far more aspects of a service and use advanced techniques such as machine learning, then the usage of annotation paths is also promising for service discovery.
We have annotated a subset of the SAWSDL-TC3[1] data-set with our annotation path method and have evaluated our matcher against service matchers that took part in the *International Semantic Service Selection Contests*[2]. We have evaluated two scenarios:

– *Scenario 1:* The goal of this scenario was to evaluate how, our simple matcher can compete against current state of the art matchers based on existing requests and advertisements of the SAWSDL-TC-3 data-set. In the first scenario we have selected one arbitrary request (*book-price*) and 40 advertisements and have annotated them manually using the annotation path method. Our matcher operated on requests and advertisements which are annotated with annotation-paths and the reference matchers used the original annotations and advertisements of the TC-3 data-set.
– *Scenario 2:* The goal of the second scenario was to asses how our matcher competes against other matchers if the semantics cannot be expressed by simple concept annotations. In this case matchers operating on simple concept annotations can only infer the missing semantics by exploiting other dimensions such as the structure or naming of elements. The second scenario was also evaluated using existing advertisements of the SAWSDL-TC3 data-set. We have only changed the request. We now require for the price of books in *Euro* but excluding tax and we restrict the input to science fiction comics. This cannot be expressed with the used ontologies because no such concepts exists. However, a hint for the standard SASWSDL matchers was provided by the requested output type (*EuroPriceExcludingVAT*) and input type (*ScienceFictionComic*).

[1] http://projects.semwebcentral.org/projects/sawsdl-tc/
[2] http://www-ags.dfki.uni-sb.de/ klusch/s3/index.html

We have executed the evaluation with the Service Matchmaker and Execution Environment (SME2[3]) which is also used for the International Semantic Service Selection Contests. Due to the partial TC3 data-set we were not able to execute all matchers. However, we could execute two major representatives iSem[3, 4] and SAWSDL-MX[5]. The *iSem* matcher when applied for SAWSDL is a hybrid service matcher exploiting inputs and outputs and service names that employs strict and approximated logical matching, text-similarity-based matching and structural matching and automatically adjusts its aggregation and ranking parameters using machine learning. It reached the best binary precision in the contest of 2012. SAWSDL-MX is a typical representative of a hybrid matcher using logics and syntax-based matching. The SAWSDL-TC3 data-set is annotated with relevance grades for each combination of advertisements and requests. A relevance grade is a value between 0 and 3, where 0 stands for not relevant and 3 stands for highly relevant. We used the original relevance grades for Scenario 1 and asked an independent expert to provide the relevance grades for Scenario 2. We have assessed the overall performance of each matcher based on the reached Normalized Discounted Cumulative Gain[2] (NDCG) which is also used in the International Semantic Service Selection Contests. The results of both scenarios are shown in Table 1. Our matcher performed more than 4 percent better than the *SAWSDL-MX* matcher and around 1 percent less precise than the nearly perfect *iSEM* matcher. In the second scenario our matcher performed around 9 percent better than *iSem* and around 12 percent better than *SAWSDL-MX*.

Table 1. Result Comparison

Matcher	NDCG Scenario 1	NDCG Senario 2
Path-Based Matcher	0.977	0.970
iSem Hybrid	0.990	0.886
SAWSDL-MX	0.937	0.867

While these preliminary results do not yet allow to draw final conclusions the annotation paths approach is promising for improving web service discovery. Our simple path-based matcher could clearly show its advantage in Scenario 2 and in Scenario 1 it could compete well with existing state of the art matchers which use far more advanced matching methods and additional aspects of service descriptions and advertisements.

5 Conclusions and Future Work

The annotation path method for semantic annotation has been developed to overcome limitations in the expressiveness of simple concept references. We showed in some feasibility tests that already a simple implementation of an

[3] http://projects.semwebcentral.org/projects/sme2/

annotation path based XML-schema matcher used for comparing web service advertisements with service requests can successfully compete with state-of the art web service discovery systems. We therefore conclude that annotation paths are well suited for capturing the semantics of objects in much finer detail and that the annotation path method and matchers based on it are promising approaches to improve web service discovery. Encouraged by the promising results we plan to evaluate our annotation path based matcher with a larger data-set and against additional existing matchers. Other future work is to integrate our matcher into existing state of the art matchers to gain even better results. Another direction of future work is to evaluate not only the matching precision but also the minimum amount of manual work to semi-automatically create annotation path annotations in comparison to simple concept references.

References

1. Dominik Joham. Path-based semantic annotation of web service descriptions for improved web service discovery. Master-thesis, AAU Klagenfurt, February 2014.
2. Jaana Kekäläinen. Binary and graded relevance in IR evaluations-Comparison of the effects on ranking of IR systems. *INFORM PROCESS MANAG*, 41(5):1019 – 1033, 2005.
3. M. Klusch and P. Kapahnke. isem: Approximated reasoning for adaptive hybrid selection of semantic services. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 184–191, Sept 2010.
4. Matthias Klusch and Patrick Kapahnke. The isem matchmaker: A flexible approach for adaptive hybrid semantic service selection. *Web Semantics: Science, Services and Agents on the World Wide Web*, 15(3), 2012.
5. Matthias Klusch, Patrick Kapahnke, and Ingo Zinnikus. Hybrid adaptive web service selection with sawsdl-mx and wsdl-analyzer. In *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 550–564. Springer Berlin Heidelberg, 2009.
6. Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Comput.*, 11(6):60–67, 2007.
7. Julius Köpke. *Declarative Semantic Annotations for XML Document Transformations and their Maintenance.* Phd-thesis, AAU Klagenfurt, March 2012.
8. Julius Köpke and Johann Eder. Semantic annotation of xml-schema for document transformations. In *Proc. of OTM'10 Workshops*, volume 6428 of *LNCS*, pages 219–228. Springer, 2010.
9. Julius Köpke and Johann Eder. Semantic invalidation of annotations due to ontology evolution. In *Proc. 2011 of OTM'2011*, volume 7045 of *LNCS*, pages 763–780. Springer, 2011.
10. Julius Köpke and Johann Eder. Logical invalidations of semantic annotations. In *Proc. of CAiSE'12*, volume 7328 of *LNCS*, pages 144–159, 2012.
11. Sabine Maßmann, Salvatore Raunich, David Aumüller, Patrick Arnold, and Erhard Rahm. Evolution of the coma match system. In *OM*, 2011.
12. Debajyoti Mukhopadhyay and Archana Chougule. A survey on web service discovery approaches. In David C. Wyld, Jan Zizka, and Dhinaharan Nagamalai, editors, *Advances in Computer Science, Engineering & Applications*, volume 166 of *Advances in Intelligent and Soft Computing*, pages 1001–1012. Springer Berlin Heidelberg, 2012.