

# Using Ontologies to Compose Transformations of XML Schema Based Documents

Johann Eder, Christian Koncilia, Marek Lehmann, Horst Pichler

University of Klagenfurt  
Dep. of Informatics-Systems  
`{firstname.lastname}@uni-klu.ac.at`

**Abstract.** We discuss how ontologies can be used to facilitate the construction of transformations between XML documents. We introduced a method to compose transformations for XML documents reusing existing (smaller) transformations to define transformations for more complex documents. Linking the types on which these transformations are built to the elements (concepts) of an ontology supports the identification of candidate transformations potentially useful for the composition. If no applicable transformations exist, the ontology can be used for identifying semantically equivalent elements and types of source and target XML-Schema.

## 1 Introduction

XML [1] has become a standard for data interchange and publication, in particular for data published on the web or transmitted through web services. Open exchange of electronic documents in XML formats frequently requires the transformation to different XML Schemas [7] or DTDs. The most widely used and popular way to transform an XML document from one schema to another is to use XSLT transformations [2]. XSLT allows to write simple transformations for small documents quite easy, but complexity and error-rates increase dramatically with the size of the documents. To define transformations the programmer requires domain knowledge and a deep understanding of the concepts being transformed. Maintenance and reuse of XSLT-code is also very difficult, therefore the creation of transformations tends to be time consuming and tedious.

We proposed to support XSLT-programmers with the concepts of the XML components and the composition of the XML transformations. We presented a solution for DTD based documents [6], which was later extended for XML Schema based documents [4, 9]. The key idea of this approach is: Given a source schema  $S$  and a target schema  $T$ , we can transform a document of  $S$  into  $T$  by decomposing the document into components for we find transformations in our transformation base and compose the transformation of the whole document from transformations of its parts. Part of this procedure is the identification of components (types, elements) of the target schema which are semantically similar to a component of the source schema.

Now we propose to link the components of both schemas to the concepts of a common ontology [5] and use this semantic relationships in the search for proper transformations. Linking the concepts of ontologies to XML components used to compose XSLT transformations will ease the production of such transformations between complex XML vocabularies.

## 2 Composition of Transformations for XML Documents

An XML component is a semantically meaningful unit of the problem domain. In an XML document the tags can be interpreted as delimiters of a component defined in such a way. In the tree model the same component is defined as a subtree identified through its root element. We extended this idea by transformations based on types defined in XML Schemas [4, 9]. The composition of XML documents allows us to write a separate transformation for every component. In our proposal the XML Schema enables us to attach a source type and a target type to XSLT transformations. A library stores information about available transformations and their relation to schemas and types and is instrumental in decomposing a document in a way that promotes the application of already available transformations on subcomponents.

XML Schema gives the programmers flexibility in defining schemas for the same documents in many different ways. There have been some efforts to define design patterns for the XML Schema [3]. In our approach the most useful pattern was called the Venetian Blind Design. In this pattern all meaningful components of an XML document (e.g. an address) are defined as named complex types, which can easily be reused in many parts of a document (e.g. elements `personalAddress` and `officeAddress` both of type `addressT` ).

## 3 Concepts in Ontologies

An ontology is a shared common understanding of a domain that can be communicated across people and computers [10]. They are generally used to build semantic-aware information systems which includes efforts to develop the so-called semantic web, medical information systems and so forth. An ontology may be represented as a set of concepts and their relationships. A concept is uniquely named within this ontology. It may be composed of other concepts, related to other concepts or be similar to other concepts. In [5] XML components were linked to concepts of an ontology. This allows to look up the exact definition of the component in the ontology, to find similar or more general or more specialized concepts and components.

## 4 Linking Ontological Concepts and XML Components

It is possible to link concepts of an ontology to XML components to support transformations between different XML vocabularies related to the same ontology. Thus the semantic description of the meaning of tags is no longer simply in

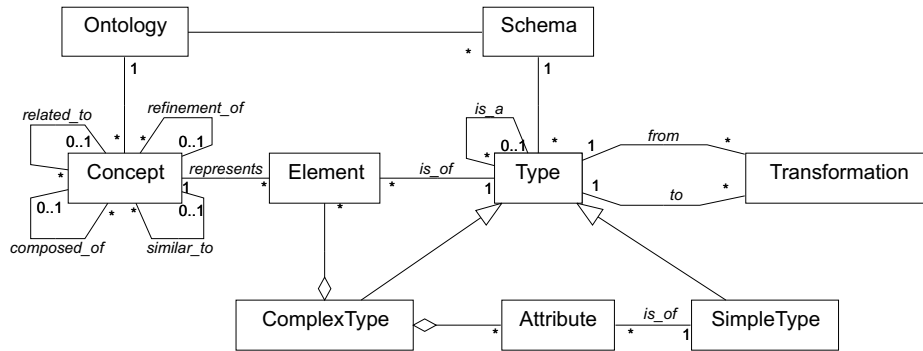


Fig. 1. Metamodel for ontological description of XML components

their name, but different named tags may represent the same concept. Some preliminary work for the DTD based documents was done in [8]. We want to extend this idea for XML Schema based documents. Fig. 1 shows a metamodel which links XML documents typed with XML Schema to concepts in an ontology.

Each XML element is of a particular XML Schema type. Types are either simple or complex. A complex type can have attributes and elements. Attributes are always of a simple type, elements can be either of a simple or a complex type. Additionally a new type can be derived from existing types.

Elements of a type can be transformed to elements of another type by applying a transformation between the source and the target XML Schema type. Each element may be related to some concept.

A concept may be composed of other concepts, related to other concepts, be similar to other concepts or be refinement of other concepts.

The ontology can be used as a data dictionary for XML Schemas and XML documents based on them, which puts the authors of transformations to a better understanding of the vocabularies being transformed.

The composition of XML transformations presented in [4] works best when all meaningful components of an XML document are defined as named complex types. Concepts related to XML elements of anonymous XML Schema types (defined locally within the element) can be identified as XML components, which enables us to attach transformations to them. This would ease the composition of transformations for XML Schemas written with a design pattern other than Venetian Blind (e.g. Russian Doll Design) [3].

The composition of XML transformation between two documents may be improved by searching the ontology for concepts related to the components of these documents and the relations between them (e.g. composition).

This extended structure is now used for decomposing an XML document and to identify suitable transformations. Given a document of schema S and the target schema T. We can decompose the document in its immediate components (elements, subtrees). Using the ontology we can search for elements and types in

the target schema which are semantically equivalent to the components of the actual document and we can search for transformations of these components. We can use the generalization hierarchy to search for transformations on more general or more specific components. If we do not find a satisfying transformation we can present a similar one to the programmer and. If there are still no matching transformations we can at least support the programmer by identifying the elements and types of the target schema which the components of the source schema should be transformed to.

## 5 Conclusions

Linking ontologies with types and elements of XML Schema descriptions can not only semantically enrich the schema definition and the metadata of XML documents but can also support the construction of transformation programs necessary for integrating independently built information systems with own document formats. Adorning the definition of XSLT programs with source and target types and registering them in a transformation library can ease the job of transformation programming. Ontologies provide the missing link between different schema definitions and allow to search for reusable transformation of components which can be composed to a transformation of the whole document.

## References

1. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler: *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation
2. J. Clark *XSL Transformations (XSLT) v1.0*. W3C Recommendation
3. K. Cagle, J. Duckett et al.: *Professional XML Schemas*. Wrox Press, 2001
4. J. Eder, M. Lehmann: *Composition of Transformations for XML Schema Based Documents*. Short paper at ADBIS 2003
5. J. Eder, M. Missikoff: *Ontologies for Managing Knowledge about Forms for Government Processes*. Proceedings of 2nd International Workshop on Knowledge Management in E-Government
6. J. Eder, W. Strametz: *Composition of XML-Transformations*. LNCS 2115
7. D. C. Fallside: *XML Schema Part 0: Primer*. W3C Recommendation
8. B. Keimel: *Application of Ontologies to Transformations of XML Documents*. Master Thesis at the University of Klagenfurt, 2002. In German.
9. M. Lehmann: *Exploiting Generalization for the Composition of Transformations of XML Schema Based Documents*. CAiSE Forum 2003, in CEUR-WS.org/Vol-74/
10. M. Uschold, M. Gruninger: *Ontologies: Principles, Methods and Applications*. Knowledge Engineering Review, Volume 11 Number 2, 1996