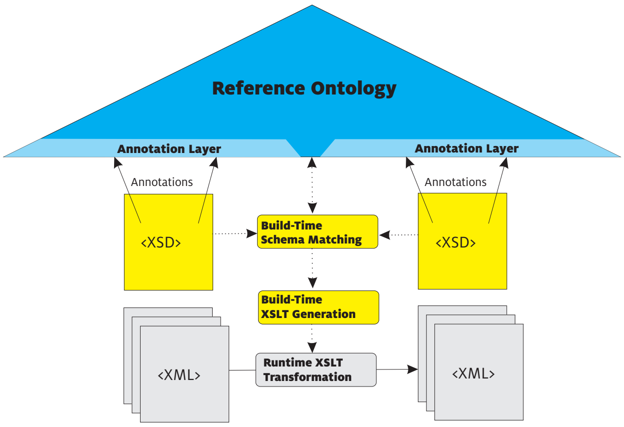
Abstract

Semantic annotations of XML-Schemas allow the interpretation of the schema elements with the help of a reference ontology. This can be a driver for the integration of heterogenous applications, where the exchanged messages needs to be transformed to the required target format.
Since an ontology conceptualizes some real-world domain and the real world constantly evolves also the reference ontology needs to evolve over time. This has consequences for the semantic annotations that need to be maintained in order to comply with the new version of the reference ontology. In addition such an evolving reference ontology gets an additional function: It does not only express the domain at some specific point in time - it can also express the evolution of the domain.
This information can be used to (semi-) automatically maintain the annotations and to detect changes that have consequences for the interpretation of instance data. This may require to change instance data in order to allow a correct interpretation with the latest ontology version.

In this thesis we will present a purely declarative annotation method for XML-Schemas, matching and mapping methods for schemas that are annotated with the proposed annotations, methods for the representation of ontology changes, methods for the maintenance of annotations with regard to structure and logics and finally the detection of schema elements, where the interpretation of the data has potentially changed.

1. Annotation-Based Transformation Approach [4]



2. Declarative Annotation Method [3]

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/ns/sawsdl" elementFormDefault="qualified" attri
  <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>
```

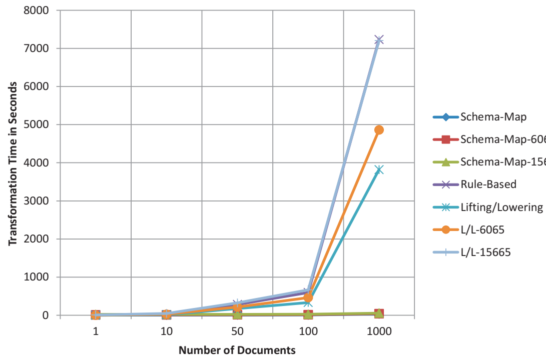
Annotation Concept

OWL-Class: /Order/billTo/Buyer/hasCountry/Country
Equivalent Classes: ConceptAnnotation and
Country and (inv) hasCountry some (Buyer and (inv) billTo some (Order))

3) Schema Matching and Mapping

Simple 1:1 matches based on semantic relations (equivalence, subclass, ...) of annotation concepts.
Complex n:m matches based on annotated transformation templates (e.g. firstname, lastname -> fullname).
*Prototype implementation g*enerates MapForce Files.

4) Performance Evaluation

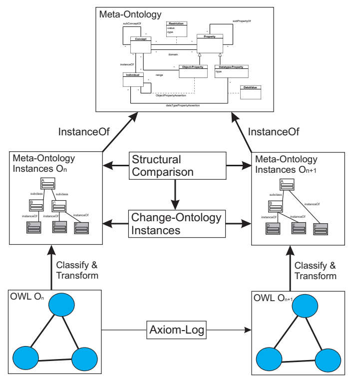


5) Change Representation

Idea: Store changes as a set of instances of a change-ontology. Each instance relates elements of the new and old ontology version with some graph operation.

Problem:
Structural and logical changes are relevelant. Metamodeling limitations of OWL. Need for tool support.

Solution:
Hybrid Meta-Ontology based approach. Can directly be queried with SPARQL.



6) Annotation Maintenance

Structural: E.g. Annotation paths reference non existing URIs. Semi-automatic *simple and complex repair strategies* based on the change representation.
Logical [1]: Systematic investigation of all types of invalidations (unsatisfiability) of annotation concepts. Definition and detection of *minimal invalid sub-paths*.
Semantical [2]: Detection of ontology changes that potentially change the interpretation of the data. Based on the explicit definition of *change-dependencies*.

Declarative Semantic Annotations for XML Document Transformations and their Maintenance

Julius Köpke
julius.koepke@aau.at

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ
FAKULTÄT FÜR TECHNISCHE WISSENSCHAFTEN
Institut für Informatik Systeme

[1] Köpke J., Eder J.: Logical Invalidations of Semantic Annotations. In: J. Ralyté, X. Franch, S. Brinkkemper, S. Wrycza (Hrsg.): Advanced Information Systems Engineering, 24th Internat. Conf., CAiSE 2012, Proc. Berlin, Springer Verlag GmbH, Juni 2012 (LNCS, 7328), pp. 144-159.

[2] Köpke J., Eder J.: Semantic Invalidation of Annotations due to Ontology Evolution. In: R. Meersman, T. Dillon, P. Herrero (Hrsg.): On the Move to Meaningful Internet Systems: OTM 2011. Springer Verlag GmbH, Oktober 2011 (LNCS, 7045), pp. 763-780.

[3] Köpke J., Eder J.: Semantic Annotation of XML-Schema for Document Transformations. In: R. Meersman, T. Dillon, P. Herrero (Hrsg.): On the Move to Meaningful Internet Systems: OTM 2010 Workshops. Springer Verlag GmbH, Oktober 2010 (LNCS, 6428), pp. 219-228.

[4] Eder J., Köpke J.: Towards Semantic Interoperability in an Evolving Environment. In: K.-D. Thoben, K. S. Pawar, B. Katzy, R. Bierwolf (Hrsg.): ICE 2009, Proc. of the 15th Internat. Conf. on Concurrent Enterprising: Collaborative Innovation: Emerging Technologies, Environments and Communities, Leiden, The Netherlands. Nottingham: CCE - (Nottingham University Business School), 2009, 4 pp.