

# XML Schemas from Computational Ontologies

R. M. Malyankar and A. Baddam

Department of Computer Science & Engineering,  
Arizona State University, Tempe, AZ 85287-5406.

[rmm@acm.org](mailto:rmm@acm.org), [anvith@asu.edu](mailto:anvith@asu.edu)

## Abstract

Academic researchers in artificial intelligence tend to use computational ontologies for knowledge modeling whereas commercial developers tend to use XML schemas, even though there is considerable overlap between the two representations considered as models of knowledge domains. Each representation has its own advantages and disadvantages compared to the other. There are active initiatives on the representation of knowledge for Web technologies, e.g., RDF (Resource Description Framework), and OWL (Web Ontology Language). These efforts often include ontology and schema editors that try to make schemas adhere to well-defined domain modeling principles, usually drawn from AI research in knowledge representation. This work is generally part of semantic web research. However, application developer communities have long accepted XML markup technology for data structuring, even though XML lacks (or rather, does not require) well-defined, rigorous domain models; the fact that such rigor is not required means that XML vocabularies are proliferating, and variant and ill-constructed models of domains and XML markup vocabularies are becoming set in stone by the effort that has gone into developing XML-based applications around them. One way to avoid this, and pave the way for future semantic web applications, is to devise a way to create and maintain a well-defined and unambiguous link between XML schemas and XML vocabularies and computational ontologies.

Our XML schema generator is intended for ontologies represented in Protege. The schema generator maps ontology classes to named *complex types* in the format of the XML Schema specification; slots in the ontology are (at the user's option) either attributes for the complex types corresponding to the classes, or sub-elements in those same complex types. Range restrictions on the values of slots in the ontology are preserved and enumerated ranges are transformed into schema enumerations. Class inheritance relationships are converted into XML schema *extension* relationships. Metadata for classes and attributes is placed into schema *annotation* elements. Schema metadata can also be entered by the user. The schema generator optionally compels adherence to the naming standards in the Department of the Navy draft guidelines for XML schema development. The validity of the XML type libraries generated by this tool has been verified using schema validation tools available on XML Schema-related web sites.

The difference between the expressive power of XML schemas and ontologies means certain features cannot be transformed. However, we find that for limited purposes, namely, turning ontologies into XML schema type libraries, the tool is useful even in its current form – it certainly speeds up the process of XML schema creation, because the taxonomical content of XML schemas, i.e., defining inheritance relationships, attributes, slot value ranges, etc., is easier with an ontology editor; so far, adding the unrepresented information to the sype libraries after the fact has not been too laborious a task – though it is certainly error-prone and repetitive, and we are investigating means of automating it or representing all the details necessary for XML schemas in ontologies.

The XML schema generator described here is being used in WIN (Waterway Information Network), a proposed distributed content management architecture for marine transportation information, described in a companion paper in this conference (Malyankar et al 2003). The rationale underlying the use of such a schema generator is outlined above and in the companion paper: facilitating linkage between AI concepts of ontologies and programming structures for XML, thereby providing a formal and logically sound basis for XML application development. We hope this is one route from current Web technology to the Semantic Web; i.e., it is transitional technology that should ease progression to the semantic web.

## Reference

(Malyankar et al. 2003): R. M. Malyankar, K. M. Shea, J. W. Spalding, M. J. Lewandowski. Managing heterogeneous models and schemas in the Waterway Information Network, DG.O 2003.