Chapter 4

Mining XML Schemas to Extract Conceptual Knowledge

Ivan Bedini  
Alcatel-Lucent Bell Labs, Ireland

Benjamin Nguyen  
University of Versailles, France

Christopher Matheus  
Alcatel-Lucent Bell Labs, Ireland

Peter F. Patel-Schneider  
Alcatel-Lucent Bell Labs, USA

Aidan Boran  
Alcatel-Lucent Bell Labs, Ireland

ABSTRACT

One of the promises of the Semantic Web is to support applications that easily and seamlessly deal with heterogeneous data. Most data in the Web, however, is in the Extensible Markup Language (XML) format, but using XML requires applications to understand the format of each data source that they access. Achieving the benefits of the Semantic Web involves transforming XML into the Semantic Web languages, OWL (the Web Ontology Language) and RDF (the Resource Description Framework), a process that generally has manual or only semi-automatic components. In this chapter, the authors present a set of patterns that enable the automatic transformation from XML Schema into RDF and OWL, enabling the direct use of much XML data in the Semantic Web. They focus on a possible logical representation of the first language and present an implementation, including a comparison with related works.

INTRODUCTION

In the last decade, the formalism of eXtensible Markup Language (XML) (Bray, 1998) has reached consensus among the most standards bodies, becoming the de facto standard format for data interchange. Several reasons motivated this choice, the first of them being that XML provides a format that is at the same time both human readable and machine interpretable. Another reason is its simplicity and suppleness of usage that fits well with the greater part of application information exchange requirements. Furthermore, the introduction of the Document Type Definition
(DTD) and XML Schema (XSD) (Fallside, 2004) formalisms permits a clean separation between meta-data and instances containing the actual data to be exchanged. Nevertheless, XML still remains, in a certain sense, too open and permits an excess of dialects that tend to overload its basic usage and meanings.

The more recent Web Ontology Language (OWL), along with the Resource Description Framework (RDF) on which it is based, has become another popular standard for data representation and exchange. Being able to translate XML Schema models to OWL ontologies through an automated process offers a significant advantage that can reduce the human work necessary when designing an ontology and the effort required to transform the Web into a Semantic Web.

Throughout this chapter we provide a pragmatic view of XML Schema practices based on a detailed analysis of Business to Business (B2B) standard specifications that, as shown in (Bedini, 2010), describes a large fraction of the usage of this technology. Our goal is to identify practical patterns for demonstrating how XML Schemas can be mined to extract ontological assertions automatically and to provide a concrete and implementable approach that improves existing systems. We show that it is not a simple process, but that this operation requires precise attention on design practices. Moreover we provide some considerations on how to best exploit the semantics given by XML Schema sources to provide labels composed by dictionary word as ontology entities names.

After this first step, we present our implementation to validate our approach and we compare the resulting data transformations with those of other systems. Indeed, as we show, some systems can already derive an OWL ontology from XML Schemas. More often the ontology is obtained with ad hoc mapping of XSD components either to OWL entities or to an intermediate data model. Rather than providing a closed set of mapping procedures, the approach we provide is based on pattern recognition. The 40 patterns we have defined are capable of mapping the most part of XSD constructions by integrating several specific design practices. This behaviour ensures a better interpretation of XML schema sources with the possibility of improving the derivation of the conceptual information handling exceptions. Our pattern-based system can also be extended simply by adding new patterns to fit other specific requirements.

This chapter is organized as follows. First, we introduce some of the main concepts on XML components, focusing on XML Schema. We continue presenting a brief analysis of XML Schema design practices based on B2B standard specifications seen as XML sources. The next sections present XML components and detail 40 transformation patterns. Then we present the prototype we have developed to validate the approach. Afterwards we provide some elements to evaluate our transformations and compare our system and approach with other systems. Finally, we conclude this chapter with a discussion of future works and research directions.

**XML DOCUMENTS AND XML SCHEMAS**

An XML Schema (Fallside, 2004) formally describes what a given XML document (Bray, 2008) contains, in the same way a database schema describes the data that can be contained in a database (tree structures, data types, integrity constraints, etc.). An XML Schema describes the coarse shape of the XML document. It can be used to express a set of rules to which an XML document must conform to be considered as ‘valid’ according to that schema, as depicted in Figure 1. Rules can define what fields or sub-element an element can contain. An XML Schema can also describe the values that can be placed into any element or attribute. At present, there exist several XML languages to describe XML documents.