XML Schema Evolution: Incremental Validation and Efficient Document Adaptation

Giovanna Guerrini[1], Marco Mesiti[2], and Matteo A. Sorrenti[2]

[1] DISI - Università di Genova, Italy
guerrini@disi.unige.it
[2] DICO – Università di Milano, Italy
mesiti@dico.unimi.it

Abstract. XML Schemas describe the structure of valid documents and can be exploited for improving both the efficiency and effectiveness of queries on valid documents. XML Schemas, however, may need to be updated to adhere to new requirements and to face changes in the application domain. Starting from a set of schema modification primitives, in this paper we devise an incremental validation approach that allows to efficiently validate documents, known to be valid for the original schema, for an updated schema. Then, we enhance the approach to adapt the documents to the new schema. Experiments prove that our approach increases the performance of standard validation algorithms in this setting and that the cost of the adaptation process is limited.

1 Introduction

XML Schemas [19] describe the structure and the allowed content of XML documents. Since the contexts where XML is exploited are highly dynamic, XML Schemas frequently need to be updated to reflect changing requirements: systems need to be adapted to real-world changes, new functionalities need to be introduced, new data types need to be processed. XML data representation formats and domain-specific schemas, before being adopted as a standard, undergo several revisions resulting in many different versions and the need arises to adapt the corresponding documents.

XML Schemas can be updated in their basic components: elements declarations, simple and complex type declarations. In [11,12] a set of primitives for evolving XML Schemas has been proposed together with an analysis of the impact of such primitives on documents known to be valid for the original schema. Documents valid for the original schema, indeed, are no longer guaranteed to meet the constraints described by the evolved schema. In principle, these documents should be *revalidated* against the new schema. A naïve approach to revalidation consists in applying a standard validation algorithm (like MSXML, Xerces, and XSV) to each document and the new schema, that has been obtained by changing the original schema through an evolution primitive. This approach, however, does not take advantage of the fact that some evolution primitives are known not to impact document validity [11,12]. Moreover, also for primitives whose application can impact validity, the evolution most likely impacts a limited portion of the schema. Consequently, only restricted portions of a document need to be revalidated. The naïve approach, moreover, does not take into account that the document is

D. Barbosa et al. (Eds.): XSym 2007, LNCS 4704, pp. 92–106, 2007.
© Springer-Verlag Berlin Heidelberg 2007

known to be valid for the original schema and that the possible effects on validity of a primitive can be foreseen. Thus, we propose in this paper an *incremental* validation approach for the validation of documents, known to be valid for an original schema sx, against an evolved schema obtained from sx through a specific evolution primitive.

If the evolution impacts validity, a related problem is how to *adapt* documents so to make them valid for the evolved schema. Adaptation by hand is error-prone and not feasible when the number of documents is high. In this paper we propose an approach in which documents are extended or pruned following a default behavior. Default adaptation is reasonable for simple evolution primitives and is very useful in some contexts, e.g., when documents are tests for statistical benchmarks.

The main contributions of this paper are an algorithm for the incremental validation of XML documents upon XML Schema evolution and an efficient algorithm for adapting the documents to the evolved schema. They have been implemented in X-Evolution [15] and experimentally evaluated. Our incremental validation algorithm outperforms the .NET validation algorithm for primitives that do not alter document validity and improves of an average 20% for other primitives. The execution time of document adaptation linearly depends on the document size.

In the remainder of the paper, Section 2 briefly surveys related work. Section 3 introduces XML Schemas and evolution primitives. Section 4 discusses validation and adaptation of complex type structures. Section 5 presents the two algorithms, that are experimentally evaluated in Section 6. Section 7 concludes the work.

2 Related Work

Schema evolution has been investigated for schemas expressed by DTDs in [14], where a set of evolution operators is proposed and discussed in detail. Problems caused by DTD evolution and the impact on existing documents are however not addressed. Moreover, since DTDs are considerably simpler than XML Schemas [5] the proposed operators do not cover all the kind of schema changes that can occur on an XML Schema. DTD evolution has also been investigated from a different perspective in [4,7]. The focus was on dynamically adapting the schema to documents. In [7] document updates invalidating some documents can lead to changes in the DTD. In [4], by contrast, modifications to the DTD are deduced by means of structure mining techniques extracting the most frequent document structures, in a context where documents are not required to exactly conform to a DTD.

In [9,18] approaches for making an XML document valid to a given DTD, by applying minimal modifications detected relying on tree edit distances, have been proposed. No knowledge of conformance of the document to a DTD is however exploited. The problem of document revalidation has been investigated in [16]. Documents to be revalidated may not be available in advance, they are known to be valid for a given schema sx_1 and must be revalidated against a different schema sx_2, but the transformations leading from sx_1 to sx_2 are not known. Incremental validation of XML documents, represented as trees, has been investigated for atomic [1,2,6] and composite [3] XML updates. Given an update operation on an XML document, the update is simulated, and only after verifying that the updated document is still valid for its schema the update is executed. An extension of the incremental validation process to document correction