

# DESIGN AND EVALUATION OF A GENERIC METHOD FOR CREATING XML SCHEMA

Mahmoud Abaza and Catherine Preston  
Athabasca University and the University of Liverpool  
mahmouda@athabascau.ca

## Abstract

*There are many tutorials and materials for learning XML syntax and how to declare elements in XML schemas. There is however no formal method to develop the schemas themselves. A consequence of this lack of method is that the development of schemas has not kept inline with the other advances in software development. The aim of this paper is to address this gap and create a schema development framework inline with both current schema development research and development methodologies. The validity of the framework was verified using case study, which ultimately demonstrated the success of this methodology.*

**Keywords:** XML schema.

## 1. Introduction

XML and its related technologies (such as XSQL, XSL, and XSLT etc) are the latest step in the development of web pages, building upon the usability of HTML and allowing, “information and services to be encoded with meaningful structure and semantics that computers and humans can understand. XML documents rely on schemas to define the structure and content allowed in each type of document. These schemas are also used by programs to ensure the content of a document is valid before it passed on to another entity such as a web service or a database for import/export.

XML has stable specification and as a result there are many tutorials and documents available on both the web and in textbooks. Much of this advice is based on the practicalities of coding a schema, that is, syntax required by a schema to make it valid both as a schema and as XML, and the actual declaration of elements. However, much fewer documents exist to provide advice and guidance on the actual processes and methods through which XML Schema can be constructed. This can therefore make it difficult to teach new authors how to create schemas as well as increasing problems for new authors, who do not have a background in programming, to write and develop meaningful and useful schemas. This lack of a standard for developing and creating schemas also means that schema creation does not meet the rigour of other programming disciplines, which have well-defined paradigms of good practice for both program design and programming. This research aims at addressing the issue of a standard method of creating XML Schemas by developing a model for constructing schemas based upon existing good software design practices and combining this with processes used by current schema writers in the field. In order to ascertain current schema design processes it will be necessary to analyze current processes by which schema are created through interviews with current schema authors as well as an analysis of good programming principles in other fields. It is proposed that the final solution should be a model of the processes by which XML schema are developed, which can be followed by any user required to construct schemas. The model developed should be generic; users from any business or industry can use it to create valid, well-formed, and useful schemas that meet the individual's specific requirements. The model developed will be evaluated through a case study.

## 2. XML Schemas

By looking at the mappings of models to schema language, it is possible to see that many of the mappings are similar even if the conceptual model on which they are based is different:

1. Reverse Engineering: Schemas can be derived from many different sources including relational databases (Fong et al., 2001) (Elmasri et al., 2002) and existing XML or other documents which may have been created prior to the development of the schema. (Collins et al., 2002; Lee et al., 2002; Fong et al., 2001) or using (Baresi and Quintarelli, 2005).
2. Forward engineering: A number of conceptual models exist which can be transformed into XML schemas. Each of these techniques will be listed below.

- Information Modelling is a technique used by Rubin et al (Rubin et al., 2002) to develop a common model for storing genetic sequence data between different study centres
- Object Oriented Modelling. In "Modelling and Transformation of Object-Orientated Conceptual Models into XML Schema" (Xiao et al., 2001).
- NAIM: Using the NAIM model as a basis for developing schemas is described by (Chankuang and Chittayasothorn, 2003).
- Object Role Modelling is a graphical model, similar to NAIM, explained by Bird, Goodchild and Halpin (Bird et al., 2000).
- UML: By far, the most common conceptual mapping technique used in the literature for developing schemas, is that of UML. (Conrad et al., 2000)

### 3.Design

In order to discover if a methodology for design , an online questionnaire was devised to ascertain the processes by which schemas were being developed in industry. The questionnaire was completed by a number of users. A review of the users responses on creating schemas shows definite similarities in the processes by which schemas are being created. These stages are: 1) System Analysis, which includes understanding business needs and analysis of data requirements, b)System Design, which includes creating rules to enforce XML contents, 3)Quality assurance, 4) Coding, and 5)Testing

These stages are similar to the four phases well-known for the phases of software design. From these, it will be possible to develop a method for creating schemas, which reflects current practice, theoretical research and existing software development paradigms. In order to discover the commonalities four tables have been constructed. Each table relates to one phase of software design and is then populated with the results discussed previously. By constructing these tables, a method for creating schemas can be developed. These tables include: a) Analysis phase, which includes understand the business need and identify schema goals; analyzing the data requirements and the requirements of the schema; Identifying existing Standards; and Identifying existing documents, B) Design Phase, which develops conceptual model C) Implementation phase, which decides schema language, namespace, logical model, physical model complex type nesting, and D) Testing phase

Now that a common set of steps has been identified, a way of presenting these for users must be found. It was decided that given its uses in other areas of software design and business processing, that UML (Unified Modelling Language) would be the best method of presenting the identified methodology. A method cannot exist alone however and so the next section presents class diagrams and use cases which aim to explain some of the concepts surrounding schemas and their development and how the method outlined above relates to the existing development of web services and uses of XML in general. Following a review of existing definitions the following class diagrams were created, outlining the classes involved in creating XML schemas and their interactions with each other as well as how XML and schemas can be used by web services and other programs (Figures 1 & 2).

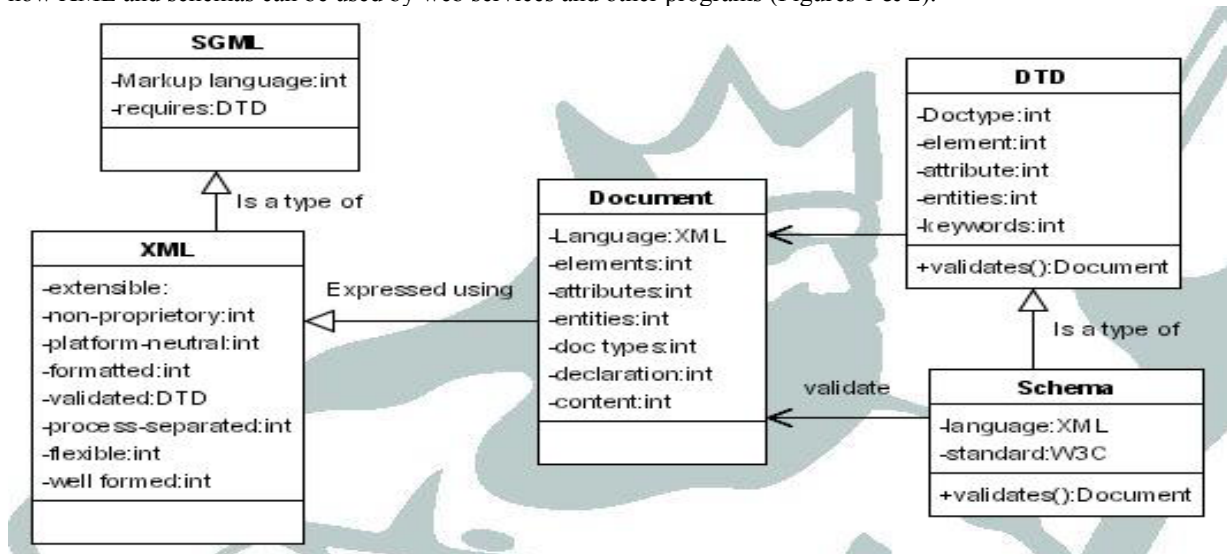


Figure 1: Class Diagram for XML

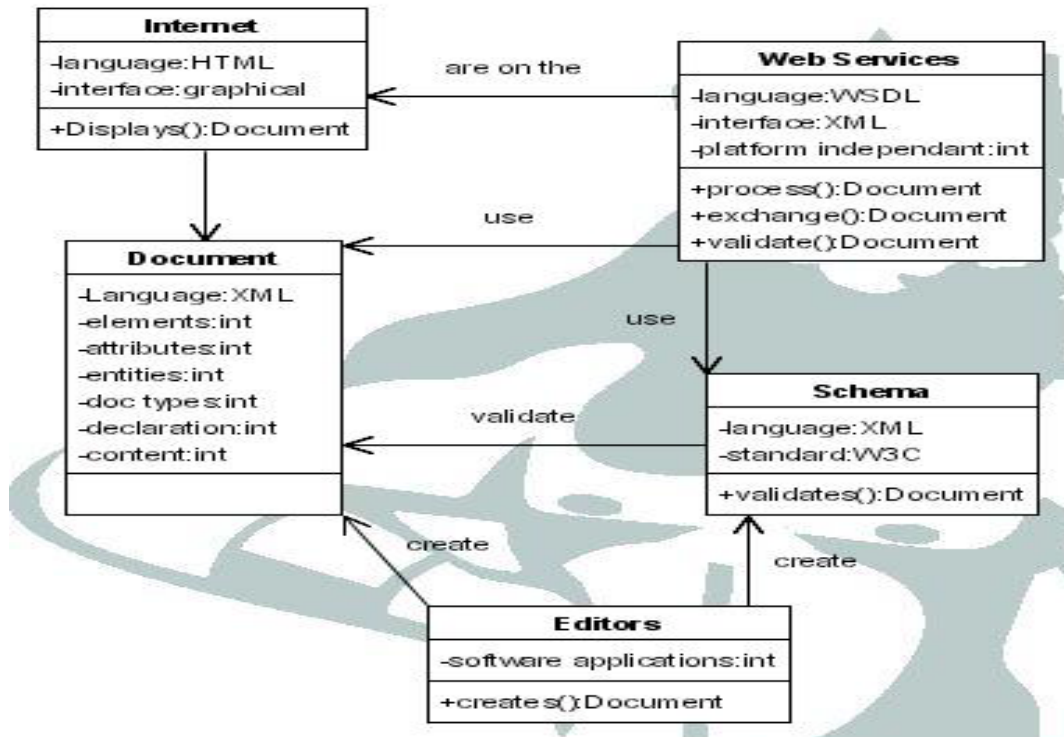


Figure 2: Using XML Documents with Web Services

Following on from the class diagram creation, cases for XML and schemas were identified. These are shown below.

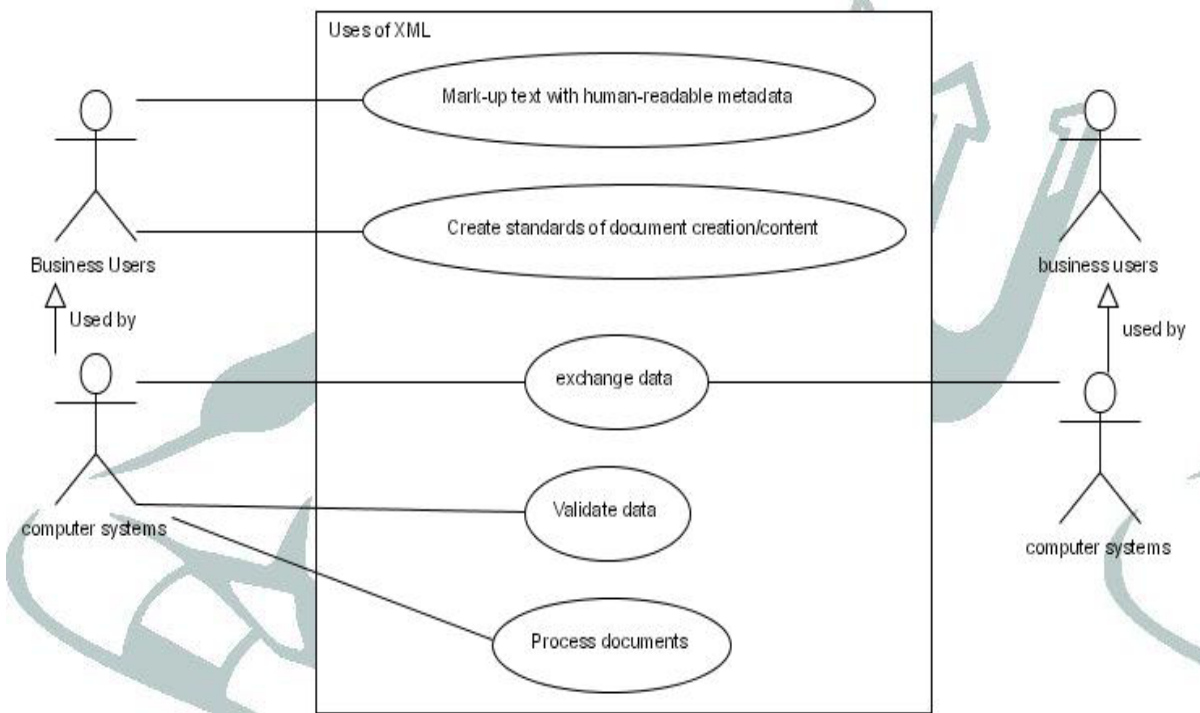


Figure 3: Business Uses of XML

## 4. Realizing the design

In order to realize the design of the methodology described in last section, activity diagrams were created for each of the four phases of software development discussed previously. Each activity diagram is based on the steps discovered in both the literature and from current creators of XML. The remainder of this section discusses the creation of each activity diagram in detail, examining the alterations made to the original method and the reasons behind these changes. As discussed earlier, the activity diagrams are to be presented using UML diagrams.

### *Analysis Phase*

Whilst creating the activity diagram for the Analysis Phase, it appeared that a reordering of the steps in this phase was needed to make more sense of the process. The new order of steps is now as follows: Understand business needs, identify schema goals, identify existing documents and standards and analyse data requirements. The end result of this phase of the schema development is the production of a schema requirement document from which the conceptual model in the design phase can be developed.

### *Design Phase*

The schema design model is based on the assumption that the users know how to choose a design method and how to model using the language chosen. This is so that the method described remains generic and easily adapted by companies or other users. As a result, the model does not detail the use of the modelling methodology although this could be added if required. Notes have been used to give some generic guidance on how models can be chosen should the user require it. The activity diagram for the design phase can be supplied by contacting the authors.

### *Implementation Phase*

In order to maintain the model's simplicity, the implementation phase has been broken down into three separate activity diagrams: Implementation, Developing the Logical Model and Developing the Physical model. This also correlates with the logical model (mapping of names of elements and attributes) and physical model (hierarchy of elements in creating schema).

This separation of the creation process allows more flexibility for authors in the development of the schema. In this method, the logical and physical models relate to mapping of the conceptual model to the schema language (coding) and then the development of the structure of the schema respectively. This allows all elements to be captured and coded before decisions about the structure and hierarchy of the schema are made, allowing for several different schemas to be created from the same conceptual model if required.

In the physical model, generalisation is removed and placed in the logical model. This is because the development of elements by restriction and extension should be done before the structure of the schema is decided. It is a natural progression of the mapping done previously. The other change to the model is the introduction of an iterative cycle for identifying elements and groupings. This change allows the users to follow the loop as many times as is necessary in a cyclical manner as opposed to the flat sequentially.

### *Testing Phase*

Although semantic testing is included in this model there are so far no definitive methods developed. Although several approaches were discussed in previous sections, there is no software available to assist in this task. Therefore this step in the process is optional. Further details could be added later as more research is undertaken in this area.

## 5. Case Study

In order to test the methodology and to ascertain how well it meets these criteria, a case study was conducted in which several schemas were created. The aim of the case study was to see not only how well the aims were achieved but also to verify the validity of the methodology developed. The case study was chosen as the evaluation method as it allows demonstrating the methodology in practice in addition to identifying areas of improvement and further development. To conduct the case study, it was necessary to establish a problem within which the methodology could be tested. This required the development of a scenario that would mimic the possible scenarios encountered by users wishing to develop schemas. In this case, a fictional company was developed and a brief was created, which would allow the methodology to be fully explored and tested. The schemas produced were valid XML schemas, verifying that the methodology enables users to create schemas successfully using standard software development practices.

## 6. Conclusion

The aim of this research was to develop a method of creating XML schemas based on current software development techniques. The result was the development of a series of steps forming the framework from which XML schemas (in a variety of schema languages) could be developed. The methodology developed is not tied to any single industry and as such could be used by a variety of users, with varying levels of knowledge on the creation of schemas and the uses of XML. The methodology could be used as a schema development framework, either by companies keen to develop their own schemas who perhaps require certain modelling techniques be used, or given by companies to external consultants to ensure that any schema created conforms to the expected industry standards. The methodology can also be applied, either in full (including the analysis and design phases) or in part (the development of a schema from existing conceptual models). The methodology could also be used as part of a program to convert existing schemas (such as DTD or even relational models) to XML XSD based schemas.

### References

- Elmasri, R., Wu, Y.-C., Hojabri, B., Li, C. and Fu, J. (2002) "Conceptual Modelling for Customized XML Schema" Lecture Notes in Computer Science (2503/2002) MetaPress [Online]
- Fong, J., Pang, F. and Bloor, C. (2001) "Converting Relational Database into XML Document", In: Proceedings: 12th International Workshop on Database and Expert Systems Applications, 2001., pp. 61 -65 IEEE .
- Baresi, L. and Quintarelli, E. (2005) "Graph Transformations to Infer Schemata from XML Documents", In: SAC '05: Proceedings of the 2005 ACM symposium on Applied computing, pp. 642 -646 ACM Digital Library [Online] Available from: <http://doi.acm.org/10.1145/1066677.1066824>
- Rubin, D. L., Shafa, F., Oliver, D. E., Hewett, M. and Altman, R. B. (2002) "Representing Genetic Sequence Data for Pharmacogenetics: an Evolutionary Approach using Ontological and Relational Models" Bioinformatics (18) Highwire Press [Online]
- Xiao, R., Dillon, T. S., Chang, E. and Feng, L. (2001) "Modelling and Transformation of Object-Orientated Conceptual Models into XML Schema" Lecture Notes in Computer Science (2113/2001) MetaPress [Online]
- Chankuang, N. and Chittayasothorn, S. (2003) "A Software Tool for Object and XML schemas Generation", In: PACRIM. 2003 IEEE Pacific Rim Conference on Communications, Computers and signal Processing, 2003, (2) pp. 675 -678 IEEE Xplore [Online] Digital Object Identifier 10.1109/PACRIM.2003.1235871