XML EVERY-FLAVOR TESTING

Antonia Bertolino, Jinghua Gao, Eda Marchetti
ISTI-CNR
Via Moruzzi 1, 56124 Pisa

Keywords: Application testing, conformance testing, DTD, XML, XML Schema.

Abstract: With XML and XML Schema widely acknowledged as the de facto standard for data exchange and interoperability between remote applications, the need for checking integrity and adequacy of XML documents, also by means of automated tools, increases. In this perspective, this paper addresses two objectives: we provide a classification and a short overview of the diverse existing approaches for the testing of XML-based documents; then, pushing further the potential of XML for testing purposes, we pursuit the application of traditional testing methods to programs using XML input data. We discuss the use of XML and XML schema as a basis for formalizing and automatizing the testing of applications using such kind of data, with particular reference to recent proposals for specification-based and perturbation-based testing approaches.

1 INTRODUCTION

In the pursuit of working interoperability among independently developed systems, industry is increasingly adopting open specifications and binding such specifications to standardised technologies. Such binding technologies must be open in nature, to allow for a wide range of diverse platforms, languages, and tools to be used, and still create compliant applications and content. The XML (eXtensible Markup Language)(W3CXML, 1996) is today the predominant format for data representation and is generally recognized as the standard way to exchange information between remote systems and to bind the specifications (W3C, 2005).

As well as the XML documents (also known as *instances*), the W3C has developed specifications for creating "control documents" that are used to define the structure of XML documents themselves. These specifications - first Document Type Definition (DTD) (DTD, 1996) and later XML Schema (XSD) (W3CXMLSchema, 1998) allow parties to construct vocabularies of tags for particular types of documents, and to insist on particular structuring rules.

Paired with XML diffusion, the DTDs and XML Schema have largely spread up. DTDs and XML Schemas are used for expressing basic structural rules and complex restrictions of the diverse data and parameters that units/components exchange with each other: both of them can be considered as the structuring schema of XML documents. The introduction of XML first, and of XML structuring schemas then, paved the way for a lot of tools and techniques devoted to check the most varied aspects and concerns of the produced documents. In fact, in parallel with the establishment of common formats for data exchange, and the standardization of parameters in communication interfaces, the need grew for ensuring the quality and integrity of the produced documents, as well as for validating that the XML products do conform, from a syntactical viewpoint, to the established standard formats and schemas, and, if feasible, also to the intended semantics. We refer broadly to the whole variety of developed technologies as "XML-based testing". However, the kind of verifications pursued, and the approaches and tools adopted vary far and wide from case to case. Besides, the various techniques are not alternative, but depending on the specific application can be combined in several interesting ways.

As is further discussed in this paper, the introduction of formalized and standardized formats for input data representation allows for a wealth of checks, and not all its potential has been explored so far. In this context, this paper focuses on two objectives:

1. For aim of clarification we provide a classification

Bertolino A., Gao J. and Marchetti E. (2006).
XML EVERY-FLAVOR TESTING.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 268-273
DOI: 10.5220/0001254602680273

of the diverse existing approaches, organized into a logical structure based on their varying types of verification.

2. From the above classification it is clear that the enormous potential of XML and similar language is only marginally exploited for testing purposes. We propose to use XML and XML Schema for ”formalizing” the testing of the applications which use such kind of data.

In the next section we overview existing approaches, distinguishing between approaches that are applied to the XML documents, and approaches that start from the XML Schema. Then, in Section 3, we look at the future, by discussing the potential of leveraging application test techniques with automated tools relying on XML input data. Finally, we draw conclusions and hint at future work.

2 A TOPOGRAPHY

The term *XML-based testing* acquires different meanings both in common sense and in literature. It can refer to verifying the adequacy of a XML document with respect to the users exigencies; verifying the adequacy of a XML instance with respect to a specific schema (DTD or XML schema); verifying the well-formedness of a schema structure; or even for defining methodologies for merging or matching diverse XML schemas.

In this vast context of diverse interpretations, this paper tries to classify the various testing approaches and represent them into a structured form (Figure 1). We identify first a course division of XML-based testing into XML documents testing and schema based testing. We provide the reader with a brief definition of what is understood for XML-based testing in the various nodes of the tree, and a tentatively complete overview of the literature.

2.1 Testing XML Documents

Over the years, XML format flexibility and its possibility to be adapted to any kind of situation increased the possibility to use it into diverse customized domains as well as for data interchange for web-sites, graphics, remote and real time applications.

With the aim of developing successful applications, which can correctly interoperate each other, verifying the correctness and adequacy of XML data becomes extremely important. For this diverse approaches have been defined as detailed in the rest of this section.

2.1.1 Well-formedness of XML Document

Due to its flexible schema, XML leaves to its users a certain freedom in writing their specific documents. With the aim of interoperability the W3C XML Core Working Group(W3C, 2005) provides a sort of core infrastructure that can be used for verifying the XML document. This is represented by a set of XML-based guidelines that provide metrics for determining the conformance to the W3C recommendations (W3C, 2005)

A first essential test that must be assessed on an XML file instance is called well-formedness (a similar test is also conducted for XML Schema, see later). This is a basic requirements for an XML file, if this property is not satisfied the tested file cannot even be classified as an XML file. Well-formedness can be easily verified, also a simple browser generally provides conformance validation features for such kind of validation, however well-formedness does not give enough guarantees on the quality of an XML file.

Using as a basis such kind of testing indications, different sets of test suites have been implemented, for instance (XMLTestSuite, 2005),(NIST, 2003). Each of them is represented by a test file (including up to diverse thousands of tests) generally associated with a test report, which contains all the background information for verifying a specific aspect of the conformance of the XML document to the basic recommendations.

2.1.2 Compliance to Specified Requirements

Along with more and more implementations using it, in order to realize the information exchange and transition between the different systems, XML documents need to conform to requirements coming from different domains. The requirements can derive from a standard or consist of some specific rules defined by the developers community, which can represent the specification of input domain or other requirements of the system.

Basically the target of the compliance are document verification and validation. Document verification is used to verify that the messages generated by the system are conforming to the input standard. Document validation instead is used to check the conformance of the content and structure of the documents, and format the documents to the requirements.

For the automatic verification and validation tools have been implemented, such as (RTTS, nd) which developed strategies for automated production of request XML documents for posting to facilitate the testing of web services and components to facilitate the validation of data content of XML documents; (XMLUnit, 2003) which enables unit testing of XML, and compare a control XML document to a test docu-

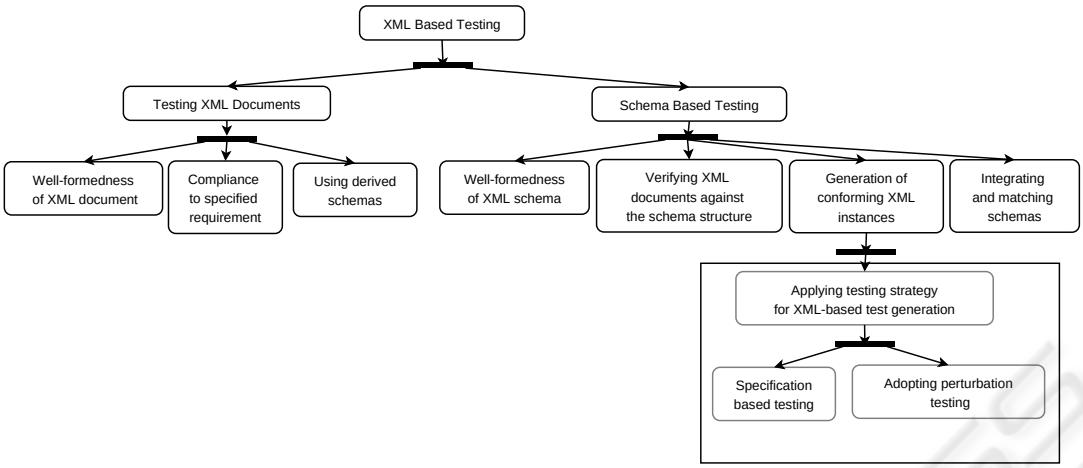

Figure 1: XML Based Testing.

ment to do the validation; (XMLTester, 2002), (XMLValidator, 2005), which use different methodologies to for large, complex systems.

2.1.3 Using Derived Schemas

When a vast amount of information must be manipulated and organized having a common reference structure is extremely important. For overcoming these problems, research on how to derive a common structure from XML instances is going on actively and interesting result have been produced (Levy, 1999), (Shanmugasundaran et al., 1999), (Widom, 1999), (Goldman and Widom, 1997). Recent applications of this technologies are (Shafazand and Tjoa, 2002), (Wang et al., 2000), which extract schema using some graphs according to the frequency of element occurrence in XML documents and (Hagen et al., 2004) in which the authors represent, by using the XML markup, a text type schema definition of the structure of the scientific paper.

In parallel with these approaches, another field of research, also called instance-level matching (Rahm and Bernstein, 2001), tries to derive a common structure by dynamically analyzing the diverse XML elements and extracting from time to time the proper schema structure. For implementing such kind of analysis diverse proposals have been adopted such as rules, neural networks, and machine learning techniques (Berlin and Motro, 2001), (Doan et al., 2000), (Doan et al., 2001), (Li and Clifton, 1994), (Li and Clifton, 2000), (Li et al., 2000).

2.2 Schema Based Testing

Many programs set their special requirements on the XML file, and can work only if the files conform to their specifications. Currently there are two solutions that are most popular to this dilemma. One is document type definition (DTD), another is XML schema, which became an official W3C recommendation in May 2001.

Establishing in fact a formalized agreement on the format of data exchange supports the application of testing strategies for checking the local data structures and the interfaces used by the different components. Using a DTD or XSD allows a recipient of an XML instance to determine whether that instance conforms to the control document by testing the XML instances for conformance against control documents. DTDs and XSDs have been the first step towards testing for the conformance of content and applications. Libraries such as Xerces provide specific interfaces for document validation and many editors are today available fort assist the XML instance developer writing instances conforming, for construction, to an XML Schema.

The recent literature collects several contributions dealing with testing of DTD and XML schema. Schema based testing can be divided into:
- well-formedness,
- Verifying XML documents against the schema structure,
- Generation of conforming XML documents from Schema.

2.2.1 Well-formedness of XML Schema

Several XML schema validators for checking the syntax and the structure of the W3C XML Schema document are available. Among them, widespread used are SQC (Schema Quality Checker) (SQC, 2001), XSV (XML Schema Validator) (W3CXMLValidator, 2001), and XML Spy5 (XMLSpy, 2005). Recently

an interesting approach has been proposed by (Li and Miller, 2005), which detects semantic errors in XML schemas by using mutation analysis.

2.2.2 Verifying XML Documents Against the Schema Structure

XML validation means checking the conformance of structure and data in an XML document against different specifications or protocols models.

Different XML documents can use different schemas (DTD or XML Schemas) to specify the valid (what they can do) and invalid (not allowed) documents. In this case a XML document can be considered valid only if everything in the document conforms to the declarations in schema. For this usually a schema should include all the elements, attributes and entities that can be used in the document as well. There exist several tools to verify the XML documents against its DTD or XML Schema. For instance, (Boobna and de Rougemont, 2004), (XMLBuddy, nd), (XMLJudge, nd) and (EasyCheXML, nd).

2.2.3 Generation of Conforming XML Instances

The widespread diffusion of XML Schema rises the proliferation of a lot of tools and methodology for deriving XML instances, which represent the allowed naming and structure of data for component interaction and for service requests. XML instances can be generated from DTD as well.

Depending on the schema, XML instances can be manually generated. This could be complicated and a huge work when schemas are complex. So tools for automated XML instance generation based on DTD or XML schema appeared. Some of them generate the XML instance directly, such as (SunXMLInstanceGenerator, 2003), (XMLGenerator, 1999), (XMLXIG, 2004), (Tian et al., 2003). Some tools generate instances in other notations, like java files (EJBSourceGenerator, 2003), (JavaXMLBindlets, 2003), C++ classes (XOMA, 2002), or .NET language such as C♯ and VB.NET (ObjectModelGenerator, 2004). But most of them generate the instances randomly. The disadvantage of random generation tools is that instances cannot cover all possibilities of the schema.

An emerging and innovative research field is thus the application of traditional testing strategies (see an overview in (Bertolino and Marchetti, 2004)) for generating suites of XML instances from the XML Schema structure as introduced in section 3.

2.2.4 Integrating and Matching Schemas

Currently the necessity for applications of managing, using and transforming diverse structure of XML data is rising. For facing this problem diverse solutions has been proposed, which can be classified depending on the level at which the integration is performed (Rahm and Bernstein, 2001).

Generally the integration of diverse schema(DTD or XML schema) can be done involving all the schemas or only part of them (Meo et al., 2005). Consider the former, diverse references can be found in literature which proposed automatic transformations safeguarding also the semantic matching of the involved schemas (Bergamaschi et al., 1999), (Doan et al., 2000), (Castano et al., 2001), (Anand and Wilde, 2005), (Jeong and Hsu, 2001), (Meo et al., 2005), (Meo et al., 2003), (Boukottaya et al., 2004).

3 XML-BASED TEST GENERATION

From a tester's point of view, the XML schema formally expresses the basic rules and complex restrictions of data and parameters that the diverse class of systems and web applications exchange, thus provide an accurate and formalized representation of the input domain. The data and parameters that system application will exchange, are in fact represented accordingly to a format suitable for automated processing, which is clearly a big advantage for testing. However so far this potentiality has been only partially exploited and the available tools only implement random generation of XML instances (sec.2.2.3). Adopting a test strategy for test case derivation will have a double positive side effect: the generation of more accurate and mindful XML instance and the improving of automatization in test cases specification. In the best of our knowledge only two works of applying a specific test strategies have been proposed which rely on the adoption of partition testing and mutation, respectively.

3.1 Specification Based Testing

The specification based techniques rely on the structural properties derived from the program specifications and different techniques can be used for guiding the section of test data ((Bertolino and Marchetti, 2004)). Considering, in particular the generation of XML instances from Schema, interesting research area is represented by equivalence partitioning. This testing strategy relies on the partitioning of the input domain into subdomains so that any input within a subdomain can be taken as a representative for the whole subset. XML Schema lends itself quite naturally to the application of equivalence partition testing. The subdivision of the input domain into subdomains can be done by exploiting the formalized representation of the XML Schema. From the diverse subdomains identified, the application of equivalence

testing amounts to the systematic derivation of a set of XML instances.

A first work in this direction is XML-based Partition Testing (XPT)(Bertolino et al., 2006). This method uses the prevalent approach to input domain partitioning, the Category Partition method (Ostrand and Balcer, 1988) combined with techniques of boundary conditions.

3.2 Adopting Perturbation Testing

In the direction of using commonly adopted testing strategies for guiding the XML based testing, another interesting work is (Offutt and Xu, 2004) that presents a new approach to testing Web services. The authors, taking as a basis the approach in (Lee and Offutt, 2001) which presents a technique for using mutation analysis to test the semantic correctness for XML based component interactions, consider communication infrastructure of web services, typically XML and SOAP, and develop new approach to testing them based on data perturbation.

4 CONCLUSION

We presented in this paper a survey of the existing approaches for the XML-based testing, trying to classifying them into a well defined structure. Our intent was twofold: first developing a reference schema that can be useful to anyone is facing with the vast and complex world of XML-Based testing. The second objective was identifying the possible field for further future researches. We in particular focus on the application of the commonly used testing strategies taking as an input the XML Schema structure. This is an innovative field of research that is only partially exploited but has a lot of potentiality also on view of automating the test cases generation. As future direction we want to investigate on this and on the basis of the works summarized in this paper, developing other innovative testing proposals. In particular we wan to focus on some new methodologies for integrating an combining the diverse existing approaches, such for instance of 2.1.3 with XPT. Finally thanks to the collaboration to diverse university and consortium working on the e-leaning environment, we want to validate our testing methodologies with case studies taken from the field.

ACKNOWLEDGMENTS

This work has been supported by the European Project TELCERT (FP6 STREP 507128).

REFERENCES

Anand, S. and Wilde, E. (2005). Mapping xml instances. Shiba, Japan. Fourteenth International World Wide Web Conference (WWW2005).

Bergamaschi, S., Castano, S., and Vincini, M. (1999). *Semantic integration of semistructured and structured data sources*, page 5459. ACM SIGMOD Record 28(1).

Berlin, J. and Motro, M. (2001). Autoplex: automated discovery of content for virtual databases. In *Proc 9th Int Conf OnCooperative Information Systems (CoopIS)*, volume 2172, page 108122, Berlin Heidelberg NewYork. Springer. Lecture Notes in Computer Science.

Bertolino, A., Gao, J., Marchetti, E., and Polini, A. (2006). Partition testing from xml schema. under submission.

Bertolino, A. and Marchetti, E. (2004). *Software Testing*, chapter 5. IEEE Computer Society. In Swebok Pierre Bourque and Robet Depuis ed.

Boobna, U. and de Rougemont, M. (2004). Correctors for xml data. In *International XML Database Symposium 2004*, pages 97–111, Toronto, Canada.

Boukottaya, A., Vanoirbeek, C., and Paganelli, F. (2004). Abou khaled: Automating xml documents transformations: a conceptual modelling based approach. volume 31 table of contents, pages 81 – 90, Dunedin, New Zealand. Proceedings of the first Asian-Pacific conference on Conceptual modelling.

Castano, S., Antonellis, V. D., and diVemercati, S. D. C. (2001). *Global viewing of heterogeneous data sources*, page 277297. IEEE Trans Data Knowl Eng 13(2).

Doan, A., Domingos, P., and Halevy, A. (2001). Reconciling schemas of disparate data sources: a machine-learning approach. In *Proc ACM SIGMOD Conf*, page 509520.

Doan, A., Domingos, P., and Levy, A. (2000). Learning source descriptions for data integration. In *ProcWebDBWorkshop*, pages 81–92.

DTD (1996). Dtd. http://www.w3.org/TR/2000/CR-SVG-20001102/svgdtd.html.

EasyCheXML (nd). Easychexml. http://www.stonebroom.com/xmlcheck.htm.

EJBSourceGenerator (2003). Ejbsourcegenerator. http://ejbgen.sourceforge.net/.

Goldman, R. and Widom, J. (1997). Dataguides: enabling query formulation and optimization in semistructured databases. In *Proc23th Int Conf On Very Large Data Bases*, page 436445.

Hagen, L., Harald, L., and Saskia, B. P. (2004). Text type structure and logical document structure. In Webber, B. and Byron, D. K., editors, *ACL 2004 Workshop on Discourse Annotation*, pages 49–56, Barcelona, Spain. Association for Computational Linguistics.

JavaXMLBindlets (2003). Javaxmlbindlets. http://wwws.sun.com/software/xml/developers /instancegenerator/index.html.

Jeong, E. and Hsu, C. (2001). *Induction of Integrated View for XML Data with Heterogeneous DTDs*, pages 151–158. CIKM 2001.

Lee, S. C. and Offutt, J. (2001). Generating test cases for xml-based web component interactions using mutation analysis. In *In Proceedings of the 12th International Symposium on Software Reliability Engineering*, pages 200–209, Hong Kong China. IEEE Computer Society Press.

Levy, A. (1999). More on data management for xml. http://www.cs.washington.edu/homes/alon/widom response.html. University of Washington.

Li, J. B. and Miller, J. (2005). *Testing the Semantics of W3C XML Schema*, pages 443 – 448. COMPSAC 2005.

Li, W. and Clifton, C. (1994). Semanticinte gration in heterogeneous databases using neural networks. In *Proc20th Int Conf On Very Large Data Bases*, page 112.

Li, W. and Clifton, C. (2000). *SemInt: a tool for identifying attribute correspondences in heterogeneous databases using neural network*, page 49 84. Data Knowl Eng 33(1).

Li, W., Clifton, C., and Liu, S. (2000). *Database integration using neural network: implementation and experiences*, page 7396. Knowl Inf Syst 2(1).

Meo, P. D., Quattrone, G., Terracina, G., and Ursino, D. (2003). Almost automatic" and semantic integration of xml schemas at various "severity" levels. pages 4–21. CoopIS/DOA/ODBASE.

Meo, P. D., Quattrone, G., Terracina, G., and Ursino, D. (2005). An approach for clustering semantically heterogeneous xml schemas. pages 329 – 346. OTM Conferences (1).

NIST (2003). Software diagnostics & conformance testing division: Web technologies. http://xw2k.sdct.itl.nist.gov/brady/xml/index.asp.

ObjectModelGenerator (2004). Objectmodelgenerator. http://sourceforge.net/projects/omgen.

Offutt, J. and Xu, W. (2004). Generating test cases for web services using data perturbation workshop on testing, analysis and verification of web services. Boston Mass.

Ostrand, T. and Balcer, M. (1988). The category-partition method for specifying and generating functional tests. *Communications of ACM*, 31(6).

Rahm, E. and Bernstein, P. A. (2001). Survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10:334–350.

RTTS (nd). Rtts: Proven xml testing strategy. http://www.rttsweb.com/services/index.cfm.

Shafazand, M. and Tjoa, A. M. (2002). *A Levelized Schema Extraction for XML Document Using User-Defined Graphs*, page 434441. Number LNCS 2510. EurAsia-ICT 2002.

Shanmugasundaran, J., Tufte, K., He, G., Zhang, C., DeWit, D., and Naughton, J. (1999). Relational databases for querying xml documents: Limitations and opportunities. In *Proceedings of the 25th VLDB Conference*.

SQC (2001). Xml schema quality checker. http://www.alphaworks.ibm.com/tech/xmlsqc.

SunXMLInstanceGenerator (2003). Sun xml instance generator. http://wwws.sun.com/software/xml/developers /instancegenerator/index.html.

Tian, K. B., Bhowmick, S. S., and Sanjay Kumar, M. (2003). *VACXENE: A User-Friendly Visual Synthetic XML Generator*. Object-Oriented and Entity-Relationship Modelling.

W3C (2005). W3c world wide web consortium. http://www.w3.org.

W3CXML (1996). W3cxml. http://www.w3.org/XML/.

W3CXMLSchema (1998). W3c xmlschema. http://www.w3.org/XML/Schema.

W3CXMLValidator (2001). W3c validator for xml schema. http://www.w3.org/2001/03/webdata/xsv.

Wang, Q., Yu, J., and Wong, K. (2000). Approximate graph schema extraction for semi-structured data. In *Proc Extending DataBase Technologies, Lecture Notes in Computer Science*, volume 1777, page 302316, Berlin Heidelberg NewYork. Springer.

Widom, J. (1999). *Data Management for XML*, volume 22(3), page 4452. IEEE Data Engineering Bulletin, Special Issue on XML. Working Document, initial draft appeared April 1999.

XMLBuddy (nd). Xmlbuddy. http://xmlbuddy.com/2.0/index.php.

XMLGenerator (1999). Xml generator. http://www.alphaworks.ibm.com/tech/xmlgenerator.

XMLJudge (nd). Xml judge. http://www.topologi.com/products/utilities/xmljudge. html.

XMLSpy (2005). Xml spy. http://www.altova.com/products _ ide.html.

XMLTester (2002). Xmltester. http://www.xmltester.org/_html_out/main/index.html.

XMLTestSuite (2005). Extensible markup language (xml) conformance test suites. http://www.w3.org/XML/Test/.

XMLUnit (2003). Xmlunit-junit and nunit testing for xml. http://xmlunit.sourceforge.net/.

XMLValidator (2005). Xml validator. http://www.elcel.com/products/xmlvalid.html.

XMLXIG (2004). Xmlxig. http://sourceforge.net/projects/xmlxig.

XOMA (2002). Xoma. http://sourceforge.net/projects/xoma.