A MODELING APPROACH FOR E-BUSINESS STANDARDS BASED ON XML SCHEMA ANNOTATIONS

Volker Schmitz
University of Duisburg-Essen, Institute for Computer Science and Business Information Systems
Universitaetsstrasse 9, 45117 Essen, Germany
volker.schmitz@uni-due.de

Joerg Leukel
University of Hohenheim, Information Systems II
Schwerzstrasse 35, 70599 Stuttgart, Germany
joerg.leukel@uni-hohenheim.de

Martin Hepp
University of Innsbruck, Digital Enterprise Research Institute (DERI)
Technikerstrasse 21a
mhepp@computer.org

ABSTRACT

Since the advent of XML as a meta language for data exchange on the Internet, numerous e-business standards for various applications and business domains have been developed. The problem is that such e-business standards can get very complex with regard to the size of respective formal specifications as well as supplementing documentations. The latter provide important information required for understanding and eventually correctly adopting such as standard. We address this problem from a document engineering perspective which provides means for specifying, designing, and implementing electronic documents. In particular, we employ the technique of XML schema annotations which allow for closely integrating both the formal specification and its supplementing documentation. We show the usefulness of our proposal by reporting on experiences made in an e-business standardization project.

KEYWORDS

Document Engineering, E-Business, Electronic Data Exchange, Standardization, XML

1. INTRODUCTION

XML formats and appropriate XML technologies are increasingly used for inter-organizational data exchange. This is reflected clearly in the high number of proposals of so called e-business ‘standards’ that support various business processes (Bussler, 2001; Söderström 2001). Extending the usage of respective XML technologies to further, more sophisticated and critical applications leads, however, to a higher complexity of the underlying XML specifications. As a result, standards adopters face large numbers of message types, data elements, and data types (Li, 200). The supplementing documentation of an e-business standard contains a lot of essential information required for understanding and eventually correctly using the standard; hence both elements – formal specification or schema and documentation – must be seen as closely inter-related. This property has also effects on both the development and maintenance as well as on the use of these standards, since each modification of the specification may require altering the documentation and vice-versa.

Specification and documentation take a central role as the results of each standardization process: The specification describes the standard formally by an XML schema language; it addresses itself primarily to machines (e.g., validators, import interfaces, and converters). The documentation, however, represents the standard for human beings. It aims at describing the semantics of all data elements in order to guarantee

correct understanding and adoption. Looking at major standardization initiatives, specifications and documentations are often developed separately from each other. One reason is the lack of adequate software tools. For instance, it is hardly possible to produce the documentation on base of the specification automatically. Commercial tools for editing XML schemas (e.g., XML Spy) include rather simple report generators. These are limited in their capabilities to produce semantically rich and customized documentations. The automatically generated documentations are confined to representing each message type's structure by briefly describing each of its items (i.e., container elements, atomic elements, XML attributes, and data types) (CEN, 2004). A closer integration and automated creation promises, on one hand, a higher result quality (e.g., enriched descriptions, completeness, avoidance of inconsistencies) and, on the other hand, a significant decrease of the time and costs for creating the documentation. This is exactly the starting point of our paper.

In this paper, we address the described problem from a document engineering perspective which provides means for specifying, designing, and implementing electronic documents (Glushko and McGrath, 2005). In particular, we employ the technique of XML schema annotations which allow for closely integrating both the formal specification and its supplementing documentation. We examine, to what extent XML schema annotations (which are already part of XML Schema, XSDL) enable an integrated development process.

The remainder of our paper is structured as follows. Next we review related work. In section 2, we identify requirements on the specification and documentation of e-business standards. In section 3, we take the document engineering perspective and look at the content of both specification and documentation by defining its structure (which information is required?), sources (where does this information come from?), and distribution (where should this information be stored?)..In section 4, we show the usefulness of our proposal by reporting on experiences made in an e-business standardization project.

Looking at research literature, the relation between specification and documentation in the context of e-business standards has been widely neglected. Besides our earlier work (Schmitz et al., 2005), most other work addresses content wise and methodical questions of the development, maintenance and adoption of e-business standards, in order to support the analysis, evaluation and selection of competitive standards; e.g., (CEN, 2004); Li, 2000; Schmitz and Leukel 2005). Another group of work deals with domain-independently XML technologies and their use, integration and extension. Apart from the fundamental XML schema languages (Lee and Chu, 2000), the integration of XML data and relational databases has to be mentioned here (Bohannon et al., 2002; Krishnamurthy et al., 2003)..Our contribution positions itself at the interface of both areas and tackles a problem of e-business standardization by reassessing the value of a somehow overseen XML technology.

2. REQUIREMENTS

E-business standards provide models for inter-organizational data exchange. It is characteristic that these models are not specific for one company, but fulfill the requirements of as much companies as possible. Otherwise they would be no standards. Therefore, the models contained in these standards can be regarded as reference models. Reference models possess a general validity for the respective domain, and are accepted as such by domain experts, i.e., companies. Therefore, we derive a set of important requirements on the specification and documentation of e-business standards from the Guidelines of Modeling (GOM) that have been proposed for the development of reference models (Schütte and Rotthowe, 1998).

Guideline of Construction Adequacy. E-business standards should be developed in such a way that the resulting models are syntactically correct and minimal. Syntactical correctness can be guaranteed by adhering to an underlying meta model. For the considered specification, this is ensured by the use of XSDL. Additionally, inconsistencies between the specification and the documentation have to be prevented. The re-use of equal concepts for different contexts should be possible. For example, the definition of message types takes place on the basis of a well-defined, complete vocabulary; this helps to build redundancy-free e-business standards.

Guideline of Language Adequacy. E-Business standards should be developed by using those languages that are suitable for the respective application and domain. Here, the duality of specification and documentation shows up: The specification must be suitable for being processed by application systems; the documentation must be suitable for its user target groups. To these groups belong, for instance, decision

makers, domain experts, system designers, and application programmers. In order to enable the subsequent treatment of the e-business standard based on the specification, it is necessary to tap the full potential of the formal specification languages as far as possible (e.g., for XSDL: domain constraints, inheritance, uniqueness, and keys).

Guideline of Economic Efficiency. E-business standards should be developed with consideration of the costs and revenues that occur during the life-cycle of the standard. In particular, established solutions for sub-problems should be taken over, if they are available; for instance, a number of ISO standards are available for coding information like languages, countries, currencies, and units of measurements. The creation of the documentation and specification as well as their customization for specific target groups (i.e., restrictions on sections of the e-business standard) should take place automatically. Moreover, it has to be considered that e-business standards are being developed spatially and temporally distributed.

Guideline of Clarity. E-business standards should be clear, unambiguous and understandable for their users. Considering that the aforementioned groups of users make different requirements on the contents, structure and form of the documentation, clarity can be regarded as the most important criteria when assessing the quality of the documentation. Its function is to describe the semantics of the standards and its models in such a way that the adoption and implementation of the standard can be established correctly and efficiently. Tools that contribute to the clarity of documentations are, for instance, navigation paths in models, naming and layout conventions, structured dictionaries of elements, glossaries, and multilingual descriptions.

Guideline of Comparability. E-business standards should be developed in such a way that they can be compared with other models of the application domain. This supports the adoption of the standard in these cases where the standard incorporates similar models or even other domain standards (e.g., mapping of its data elements to others). However, the guideline of comparability can be in conflict with the business model and the strategic goals of the standardization organization, since the comparability could favor or enable the move to other competitive standards. Comparability is supported, for instance, by version information (comparability with prior versions of the same standard), cross-standard vocabularies (e.g., ebXML core components), standardized modeling languages (e.g., UML, XSDL) and methodologies (e.g., UMM).

Guideline of Systematic Design. E-business standards should be specified by adhering to a meta model that integrates relevant views and levels of modeling. This will guarantee the syntactic correctness and consistency of all specifications. Since the standards regarded here are based on XML technologies, the meta model is already supplied by XSDL. For this meta model, three levels of e-business standardization are important (data view): message types, elements (tags and attributes), and data types. The respective super ordinate level supports the re-use of the definitions of the subordinated level. In the context of e-business standardization another level has to be mentioned: The process level describes the sequence of exchanged business documents and the underlying business logic (process view) (Schmitz and Leukel, 2005). It is, however, not covered by the XML meta model.

3. DOCUMENT ENGINEERING PROCESS

The basic idea point of our proposal is an integrated view of specification and documentation: Both are essential components of each e-business standard; however, they address different subjects (machines vs. humans). Therefore, their development should be integrated in the document engineering process.

3.1 Document Content

A central part of the documentation of each e-business standard is the description of its vocabulary. Its items are described, for instance, by its data type, sub elements, and attributes. Table 1 shows respective meta data for XML elements and attributes.

Table 1. Meta data for XML elements and attributes

Meta data	Content
Name (formal)	Name of the item as in the XML schema (e.g., PRODUCTID)
Name (textual)	Written out name of the item (e.g., Product Identifier)
Short description	Short description
Long description	Detailed description; may include illustrations or tables
- Examples	Examples that explain the usage (XML code)
- Changes per version	Textual description of the changes due to new versions
Graphical representation	Graphical representation of the item and its sub items (e.g., elements with sub elements and attributes)
Data type	Data type of the item
Data type facet	Facet of the data type, i.e., field length or minimum/maximum values
- Values (permitted)	Domain restriction with permitted values (enumeration)
- Values (predefined)	Domain restriction with predefined values (pattern)
Default value	Default value of the item
Language dependency	Indication whether the content of the item depends on the used language and therefore can be specified multiple
Usage (super ordinate elements)	List of elements where the item is used
Sub elements	List of sub elements of the item
Version number	Number of the version in which the item was changed last
XSD / DTD extract	Extract of the XML schema with the definition of the item
Cardinalities	Cardinalities of the sub elements
Attributes	List of attributes of the element

3.2 Sources of Information

For guaranteeing the consistency of specification and documentation as well as for the automated creation of the documentation based on the specification, all relevant information has to be represented in such a way that it is electronically processible.

Table 2. Mapping of meta data to item types (4) and sources of information (3)

Meta data	Element	Attribute	Data type	Value
Name (formal)				
Name (textual)				
Short description				
Long description				
- Examples				
- Changes per version				
Graphical representation				
Data type				
Data type facets				
- Values (permitted)				
- Values (predefined)				
Default value				
Language dependency				
Usage (super ordinate elements)				
Version number				
XSD / DTD extract				
Sub elements (see element)				
Cardinalities				
Attributes				

We categorize the information according to its source as follows (see table 2):

- The darkly marked information is already contained in the XML schema of the standard. A prerequisite is, however, that the appropriate modeling concepts of XSDL are actually used. Our survey of prominent e-business standards showed that this does not take place to the full degree (Schmitz et al. 2003).
- The medium marked information can be derived from the XML schema, if modeling conventions are followed. For instance, if constantly a special data type or a special attribute is used for specifying the language of an element, then we can derive that this element is language-dependent.
- The brightly marked information can not be extracted from the XML schema; hence this information has to be supplied additionally. Representing all this information in an XML format is feasible due to two reasons: First, this information is semi-structured (i.e., textual descriptions). Second, its subsequent processing together with the XML schema is much simpler, since both are coded in XML. In particular, data transformations can be formulated with XSLT[1].
- The non-marked information is not relevant for the respective item type (e.g., attributes do not have attributes).

From the requirements of reusability (guidelines of economic efficiency and systematic design) and distributed development (guideline of economic efficiency) follow the need for storing the information specified above as separately as possible in independent files. These files can be re-united automatically, if necessary. The distribution of the information takes place dependent on the scope; e.g., if the information concerns an element and the information is valid for all contexts of this element, then the information can be directly added to the specification of the respective element. We distinguish the following groups of information (see figure 1):

Element-specific information serves for the description of the elements (and data types) being part of the vocabulary.

Context-independent element-specific information describes characteristics that are independent from the use of the element (context, scope); for instance, the element name must not change at all.

Context-dependent element-specific information describes characteristics that are only valid in its context, thus they may be different in other contexts. The context is determined by the super ordinate element to which the element is assigned. For example, it can be expressed by a context-dependent long description that the PICTURE element in the context of SUPPLIER represents the supplier's logo, while it serves in the context of PRODUCT for product figures.

Element-spanning information refers to the standard generally or to components of it (e.g., introduction texts, explanations concerning the structure of the documentation, legal notes etc.).

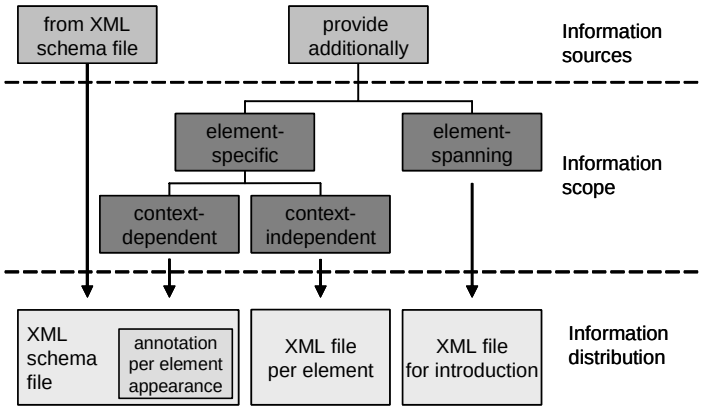

Figure 1. Sources of information, ranges of validity and distribution

[1] XSLT 2.0: XSL Transformations (XSLT) Version 2.0; http://www.w3.org

3.3 Distribution of Information

Figure 1 has already shown that the element-spanning information is stored in one XML file while the context-independent information is stored in a single XML document per each element. All context-dependent information is bound to the structure specified in the XML schema. Therefore, it is not meaningful to build a second, parallel structure. Thus the context-dependent information can be inserted directly into the XML schema. Such an XML schema, which has been enriched by semantic information, is called “annotated schema”.

There are two alternatives for enriching XML schemas: On one hand, the already defined elements can be extended by additional XML attributes of a new name space. On the other hand, the “xsd:annotation” can be used which is explicitly intended for providing documentation and application information within XML schemas. We choose the latter – "xsd:annotation" – because it draws a clear dividing line to the ordinary "xsd:element"s.

Since most meta data is semi-structured, we use a subset of X-HTML for coding this information (e.g., formatted texts, lists, figures, and tables). This subset has been determined in a way that even different output formats will result in similar results (i.e., PDF, DOC, and HTML). Additionally, some new tags with special semantics had to be defined in order to build, for instance, sections and ordered headings, insert tables of contents and indices, format XML fragments, place pictograms, and emphasize important comments. Moreover, internal references can be set in the text, e.g., to sections, elements, attributes, values, data types, and examples; these references can be checked automatically whether they are valid or not.

4. EVALUATION

In this section, we describe the application of our proposal in a real-world e-business standardization project and report on experiences made. We implemented a software prototype for the standardization projects QDX 1.0 and BMEcat 2005. QDX (Quality Data Exchange) is a standard for the exchange of quality data between suppliers and manufacturers in the automotive industry (www.vda-qmc.org). BMEcat 2005 is the current version of the BMEcat standard for the exchange of electronic product catalogs in B2B e-commerce (www.bmecat.org). In the following we refer only to the BMEcat project and compare the development and maintenance phases of version 1.2 (manual documentation process) with version 2005 for which we automated the documentation process). The results and experiences span the work from the further development of BMEcat version 1.2, over the maintenance of version 1.2 to the current version 2005. Table 3 compares both versions based on quantitative criteria reflecting the size of their specifications and documentations.

Table 3. Comparison of BMEcat 1.2 and BMEcat 2005

Criteria	**BMEcat 1.2**	**BMEcat 2005**
Messages types	3	3
Size: Number of Elements	182	401
Size: Lines of Code	4,480	8,120
Size: Number of Documentations	2	5
Size: Pages of all Documentations	185	697
Output Formats of Documentations	1	3

The BMEcat 2005 documentation consists of a base document and four additional, module-specific documents. Three file formats are supported: PDF, HTML, and compiled windows help file (CHM). All documents have the same structure: cover page, legal notes, table of contents, introduction with subsections, dictionary of elements (with graphical representations of the XML document structure), index page (only PDF), dictionary of data types, list of modifications, alphabetical list of elements, and table of contents as bookmark .

All documents are generated automatically. Figure 2 describes the generation process and shows the involved file types and used software tools. The generation is realized using XSLT scripts which are

processed by SAXON[2]. The HTML pages are produced directly, the diagrams via the intermediate XML-based format SVG[3] with Batik[4], and the PDF files with XSL-FO[5] and FOP[6], which is a Java application for processing XSL formatting objects.

The batch processing is controlled by the Java-based built tool Ant[7] and a parameter file. With this file all adjustments, like selection of the target language, restrictions of the document coverage, structure of the modules, and definition of stylesheets are specified. The BMEcat XML schema files are published without the schema annotations. Therefore, all annotations are removed from the XSD files prior to publication.

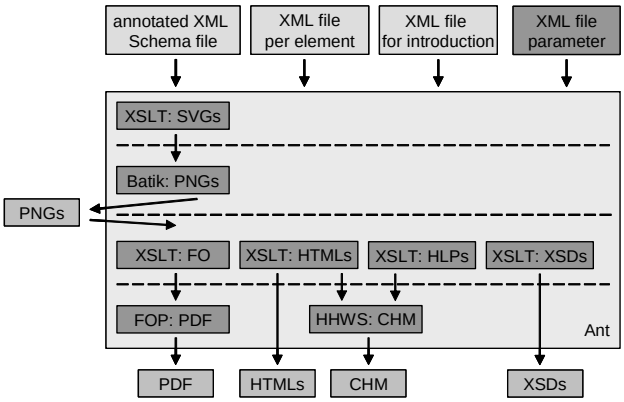

Figure 2. Steps for the automatic creation of the documentation

The concept and its implementation worked in the field use. The following improvements were achieved during the standardization process (compared to the previous BMEcat version):

- The time, which had to be invested into quality assurance, could be significantly reduced, since by using the automatic generation of the documentation on base of the annotated XML schema inconsistencies between the formal specification and the non-formal documentation could be prevented in advance (see also empirical data in table 4).
- The developers of the standard could concentrate on the technical specification and the content wise description, since manual, page-oriented editing and formatting were omitted (in particular of tables and directories). In BMEcat 1.2 all documentations were edited with MS Word.
- An additional efficiency increase could be obtained by generating reports; these contain internal comments, incorrect references, and missing texts. Reports supported also the steering of the project, since the completion degree could be assessed. In addition, this will be used in the future for the localization of BMEcat (creating the English documentation on base of the German version).

The highly distributed data storage of all information (annotated XML schemas, several hundred XML documents for all items) supported the creation of documentations that describe specific aspects of the entire BMEcat vocabulary only. Furthermore, it was possible that at the same time several developers could work on the documentation. Finally, during the QDX development, we could reuse information that was already coded for BMEcat.

The concept supports in particular the late phases of the standardization process. By automation it was possible to generate intermediate versions fast and to establish a continuous alignment between the work of the developers and the evaluation by domain experts. The first responses from the users of the standard regarding the new documentations are likewise positive.

[2] Saxon 7.9.1: SAXON - The XSLT and XQuery Processor; http://saxon.sourceforge.net
[3] SVG: Scalable Vector Graphics (SVG) 1.1 Specification; http://www.w3.org/Graphics/SVG
[4] Batik 1.5.1: Batik SVG Toolkit; http://xml.apache.org/batik
[5] XSL-Fo 1.0: Extensible Stylesheet Language (XSL) Version 1.0; http://www.w3.org/TR/xsl
[6] FOP 0.20.5: Formatting Objects Processor; http://xml.apache.org/fop
[7] Ant 1.6: Apache Ant 1.6; http://ant.apache.org

Table 4. Experiences with BMEcat 1.2 and BMEcat 2005

Criteria	**BMEcat 1.2**	**BMEcat 2005**
Average time for updates		
- Create element	45 min.	2 min.
- Restructure element	15 min.	2 min.
- Add (simple) element description	20 min.	5 min.
- Delete element	10 min.	2 min.
Time for structural quality assurance	40 h	0 h
Structural errors after publication	8	0

At present the generator has still the character of a tool box. Thus, graphical user interfaces for editing the documentation and controlling the generation process are missing. Therefore, in the current development stage experts of the standardization domain can not work on the documentation on their own and are dependent on the support of the developers that must have knowledge of XML, X-HTML and XSLT.

5. CONCLUSION

This paper addressed the problem of increasingly complex e-business standards as reflected in its two closely related elements, formal specification and documentation. We addressed this problem from a document engineering perspective which yields means for specifying, designing, and implementing electronic documents. We have proposed using XML schema annotations as a technique for linking specification and documentation and thus integrating their content on a formal basis.

Our first experiences made in a real-world standardization project showed the usefulness of our proposal. The resulting XML schema contains all relevant information needed for generating a user friendly and adequate documentation directly and solely based on the XML schema and additional machine readable files. The results show evidence that our proposal helps to improve both the development and maintenance of e-business standards by speeding up the processes and avoiding errors and inconsistencies with the help of annotated schemas.

REFERENCES

Bohannon, P .et al, 2002. From XML Schema to Relations: A Cost-Based Approach to XML Storage. *Proceedings of the 18th International Conference on Data Engineering (ICDE.*, San Jose, California, USA, pp. 64-75.

Bussler, C., 2001: B2B Protocol Standards and their Role in Semantic B2B Integration Engines. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,* Vol. 24, pp. 3-11.

CEN, 2004.CWA 15045:2004 E – Multilingual catalogue strategies for eCommerce and eBusiness, Brussels, Belgium.

Glushko, R.J., McGrath, T., 2005. *Document Engineering: Analyzing and Designing Documents for Business Informatics and Web Services*. MIT Press, Cambridge.

Krishnamurthy, R. et al, 2003. XML-SQL query translation literature: The state of the art and open problems. *Proceedings of XSym*, Berlin, Germany, pp. 1-18.

Lee, D., Chu, W.W., 2000. Comparative Analysis of Six XML Schema Languages. *ACM SIGMOD Record*, Vol. 29, pp. 76-87.

Li, H., 2000. XML and Industrial Standards for Electronic Commerce. *Knowledge and Information Systems,* Vol. 2, pp. 487-497.

Schmitz, V. et al, 2003. Does B2B Data Exchange tap the full Potential of XML Schema Languages. *Proceedings of the 16th Bled Electronic Commerce Conference*, Bled, Slovenia, pp. 172-182.

Schmitz, V., Leukel, J., 2005. Findings and Recommendations from a Pan-European Research Project: Comparative Analysis of E-Catalog Standards. *International Journal of IT Standards and Standardization Research (JITSR),* Vol. 3, pp. 51-65.

Schmitz, V., Leukel, J., Hepp, M., 2005. Integrierte Dokumentation und Spezifikation von E-Business-Standards mit XML Schema-Annotationen. *Proceedings Berliner XML Tage 2005 (BXML 2005)*, Berlin, pp. 179-190.

Schütte, R., Rotthowe, T., 1998. The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models. *Proceedings of the 17th International Conference on Conceptual Modeling (ER '98),* Singapore, pp. 240-254.

Söderström, E., 2001. The Role of Standardisation in Inter-Organisational Business Processes. *Proceedings of the 2001 IEEE Conference on Standardization and Innovation in Information Technology*, Boulder, USA, pp. 263-271