Integrated Specification and Documentation based on XML Schema Annotations

Valter Schmitz, Joerg Leukel, University of Duisburg-Essen, Institute for Computer Science and Business Information Systems, Essen

Numerous XML-based formats have been developed to realize intra- and inter-organizational data exchange. This is especially true for e-business standards that support business processes. Initially started as small XML messages covering only very specific transactions, today often extensive sets of message types are developed. Extending the usage of XML technologies to further, more sophisticated and critical applications leads, however, to a higher complexity of the underlying XML formats. As a result, standards adopters face large numbers of message types, data elements and data types. Providing an unambiguous, consistent and user-friendly documentation in an efficient way is the major challenge in e-business standardization. One of the main tasks is to ensure that the formal specification specified in XML Schema (XSDL) and the textual documentation are consistent.

Next we present a concept for enriching XML schemas with annotations to combine the formal specification with the textual description. This approach allows the generation of the documentation fully automated and without inconsistencies. Looking at major standardization initiatives, specification and documentation are often developed separately from each other. One reason is the lack of powerful software tools for engineering XML message types and producing customized documentation.

Information Requirements

The central part of the documentation of is the description of the vocabulary. Its items are described, for instance, by its data type, sub elements, and attributes. Figure 1 shows respective meta data for XML elements and attributes.
<table>
<thead>
<tr>
<th>Meta data</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name (formal)</td>
<td>Name of the item as in the XML schema (e.g., PROJECT)</td>
</tr>
<tr>
<td>Name (textual)</td>
<td>Written out name of the item (e.g., Product Identifier)</td>
</tr>
<tr>
<td>Short description</td>
<td>Short description</td>
</tr>
<tr>
<td>Long description</td>
<td>Detailed description; may include illustrations of tables</td>
</tr>
<tr>
<td>– Examples</td>
<td>Examples that explain the usage (XML code)</td>
</tr>
<tr>
<td>– Changes per version</td>
<td>Textual description of the changes due to new versions</td>
</tr>
<tr>
<td>Graphical representation</td>
<td>Graphical representation of the item and its sub items (e.g., elements with sub elements and attributes)</td>
</tr>
<tr>
<td>Data type</td>
<td>Data type of the item</td>
</tr>
<tr>
<td>Data type facet</td>
<td>Facet of the data type, i.e., field length or minimum/maximum values</td>
</tr>
<tr>
<td>– Values (unlimited)</td>
<td>Domain restriction with permitted values (enumeration)</td>
</tr>
<tr>
<td>– Values (predefined)</td>
<td>Domain restriction with predefined values (pattern)</td>
</tr>
<tr>
<td>Default value</td>
<td>Default value of the item</td>
</tr>
<tr>
<td>Language dependency</td>
<td>Indication whether the content of the item depends on the used language and therefore can be specified multiple times</td>
</tr>
<tr>
<td>Usage (superordinate elements)</td>
<td>List of elements where the item is used</td>
</tr>
<tr>
<td>Sub elements</td>
<td>List of sub elements of the item</td>
</tr>
<tr>
<td>Version number</td>
<td>Number of the version in which the item was changed last</td>
</tr>
<tr>
<td>XSD / DTD extract</td>
<td>Extract of the XML schema with the definition of the item</td>
</tr>
<tr>
<td>Cardinalities</td>
<td>Cardinalities of the sub elements</td>
</tr>
<tr>
<td>Attributes</td>
<td>List of attributes of the element</td>
</tr>
</tbody>
</table>

Fig. 1: Meta data for XML elements and attributes

Information Sources
For generating the documentation based on the specification automatically, all relevant information has to be represented machine-readable. We categorize this information as follows (see Figure 2):

- The darkly marked information is already contained in the XML schema of the standard. A prerequisite is, however, that the respective modeling concepts of XSDL are actually used.
- The medium marked information can be derived from the XML schema. Hence, this information has to be supplied additionally.
- The non-marked information is not relevant for the respective item type (e.g., attributes do not have attributes).
<table>
<thead>
<tr>
<th>Meta data</th>
<th>Element</th>
<th>Attribute</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name (human)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name (technical)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long description</td>
<td>- Examples</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Changes per version</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical representation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data type facets</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Values (permitted)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Values (predefined)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language dependency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage (super ordinate elements)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XSD / DTD extract</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub elements (see element)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Information Distribution

From the requirements of reusability and distributed development follows the need for storing the information specified above as separately as possible in independent files. The distribution of the information takes place dependent on the scope. We distinguish the following groups of information (see figure 3):

- 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.
- Element-spanning information refers to the standard generally or to components of it (e.g., introduction, legal notes etc.).
Schema Annotations

Figure 3 has already shown that the element-spanning information as well as the context-independent information is stored in a single XML document per each element. All context-dependent is bound to the structure specified in the XML schema. Therefore, it is not meaningful to build a second, parallel structure. Thus 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 because it draws a clear dividing line to the ordinary "xsd:element"s.

Evaluation

Our concept has been evaluated by implementing a respective software tool and using this tool for the standardization project BMEcat 2005, which is the newest version of the BMEcat standard for the exchange of electronic product catalogs in B2B e-commerce (www.bmecat.org). The BMEcat vocabulary consists of 401 elements, the documentation totals 697 pages.

Figure 5 describes the generation process and shows the involved file types and used software tools.

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 inconsistencies between the formal specification and the non-formal documentation were prevented in advance.
- The developers could concentrate on the technical specification and the content wise description, since manual, page-oriented editing and
Fig. 4: BMEcat documentation

Fig. 5: Steps for the automatic creation of the documentation

formatting were omitted (in particular of tables and directories). In BMEcat 1.2 all documentations were edited with MS Word.

- Internal reports listing comments, incorrect references, and missing texts 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 distributed storage of all information (annotated XML-schemas, several hundred XML files for all items) enables the creation of documents 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.

Contact: volker.schnatz@uni-essen.de