ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LVIII 29 Number 6, 2010

PRINCIPLES OF REUSABILITY OF XML-BASED ENTERPRISE DOCUMENTS

R. Malo

Received: August 31, 2010

Abstract

MALO, R.: *Principles of reusability of XML-based enterprise documents.* Acta univ. agric. et silvic. Mendel. Brun., 2010, LVIII, No. 6, pp. 295–302

XML (Extensible Markup Language) represents one of flexible platforms for processing enterprise documents. Its simple syntax and powerful software infrastructure for processing this type of documents is a guarantee for high interoperability of individual documents. XML is today one of technologies influencing all aspects of ICT area.

In the paper questions and basic principles of reusing XML-based documents are described in the field of enterprise documents. If we use XML databases or XML data types for storing these types of documents then partial redundancy could be expected due to possible documents' similarity. This similarity can be found especially in documents' structure and also in documents' content and its elimination is necessary part of data optimization.

The main idea of the paper is focused to possibilities how to think about dividing complex XML documents into independent fragments that can be used as standalone documents and how to process them.

Conclusions could be applied within software tools working with XML-based structured data and documents as document management systems or content management systems.

XML, XML modularization, enterprise documents

A document is a collection of pieces of text (character strings) organized according to a specific structure. In order to make a document's structure explicit, additional information must be interspersed among the natural text of the document (Ramalho, Almeida and Henriques; 1997).

In fact, this citation is the essence of markup languages as XML or SGML. These markup languages are designed to offer a mechanism for specifying logical elements in documents by the simple way. And this is very important possibility especially for enterprises and also for automated enterprise information systems that are responsible for processing various types of documents and their mediation to users. That is why especially XML-based languages are understood today as a significant platform for document management system and electronic data exchange.

In enterprise environment many types of documents are used and majority of them represents structural data. It means it is possible to describe explicitly their mostly unified structure. XML is only one of ways how to do it, but there is not another so powerful tool, advantages are clear (see Yao, Trappey, Ho, P.; 2003) Main representatives of enterprise documents with unified structure are for example:

- invoice,
- business proposal,
- order,
- application form,
- letters

and many other with relations to enterprise business. Out of business fields enterprises produce also many documents in internal communications with employees as rules, documentations, study texts, or circular notes. Because of high uniformity of stated documents a set of templates can be used for their preparation. In simple way a template is a special empty document with delimitated logical parts that should be filled by authors. Templates provide easy author tool and also basic check-out of results. The whole set of them is usually necessary part of document management systems used in firms and other institutions.

Sometimes two or more physical documents are almost the same or one of them is a part of the other or simply one part of the first document is identical to any part of the others. And this is a good reason for talking about reusability of documents.

Reusability of documents itself can be characterized along three dimensions (Boll, Klas, Westermann; 2004):

- granularity of reuse;
- kind of reuse;
- support for identification of reusable components.

The granularity of reuse determines what can be reused and represents the very important question how to set parts of document to be reused. Regarding common documents we can distinguish three basic levels of granularity in documents:

- complete document;
- logical parts (chapters, sections etc.);
- logical elements in document (paragraph, table or multimedia elements).

But the whole described problem is not so easy. There are two significant partial problems to be solved.

The first of them is a problem how to set which parts of document or document itself could be signed as reusable components. It means how to delimitate modules of documents with potential reusability. It is necessary to note that enterprises' documents could be mostly reused only at the level of complete document or top logical parts. The lower levels are so detail that possible real utilization is probable very low.

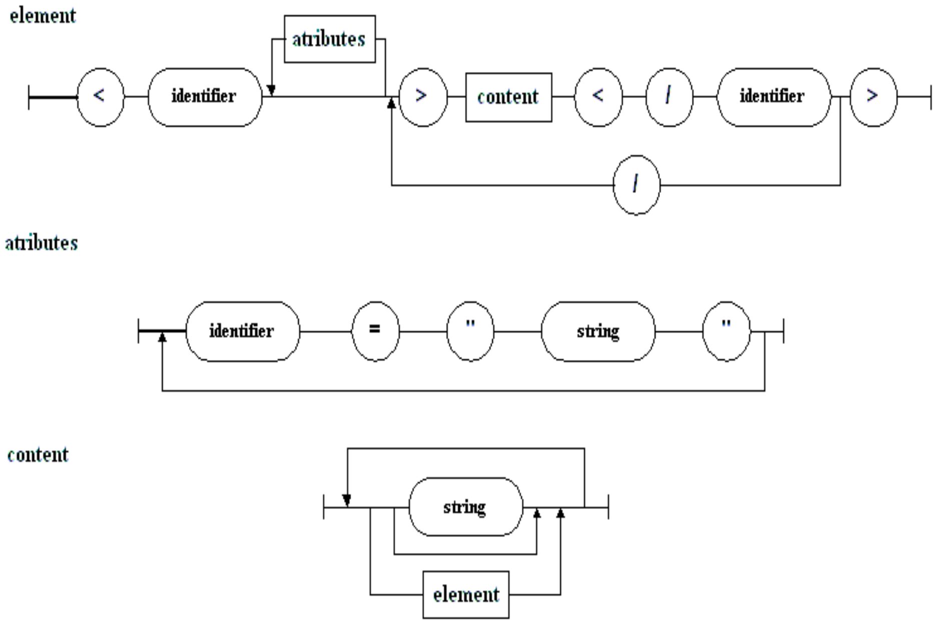
The second problem is how to use all defined (and separated) parts together in one basic document. In this case the solution can be found in the principle of linking partial XML components into one main document.

MATERIAL AND METHODS

The important outlets of this work are identification and specification of basic concepts and facts. The base is the essence of using XML documents and principle of document modularization necessary for the determination of suitable approaches supporting reusability. In next text a brief description of XML documents and DTD is provided including principle of document modularization.

XML Based Documents

XML (Extensible Markup Language) is metamarkup language suitable for a definition of a structure of structured and semi-structured data. Language itself is based upon SGML and that is why DTD (Document Type Definition) may be used as a mechanism for formalizing given structure.[1] But in



1: *XML element, attribute and content syntax diagram*

[1] There are also other sophisticated languages with the same goals (XML Schema, NG Relax, Schematron). Principles are the same and that is why only DTD is considered in this paper.

fact, it is possible to use XML also without explicitly defined structure.

Example XML document representing selected data from simple invoice could be written as follows:

```
<Invoice>
  <Invoice_info>
    <Invoice_number>PL876501234</Invoice_number>
    <Date>2007-06-13</Date>
    …
  </Invoice_info>
  <Supplier>
    <Name>Example Company</Name>
    <Address>
         <Street>Nothern street</Street>
         <Number>1</Number>
         ...
    </Adress>
    <Tel>+420 1 2345 6789</Tel>
  </Supplier>
</Invoice>
```

Syntax of XML document can be simply described by the syntax diagram in fig. 1.

Every XML document is principally standalone, but when using such structure for example as a format for electronic data interchange or as a template, its structure should be unified a strictly described by special schema (DTD is the simplest variant). In DTD all declarations of elements (ELEMENT) and also attributes (ATTLIST) have to be written strictly in given syntax similar to Backus-Naur syntax. In fact, DTD (or another schema) represents a template for one type of XML documents with delimited mandatory and optional parts.

```
<!ELEMENT  element_name (element_content)>
<!ATTLIST  element_name
           att1_name att1_type att1_cardinality
           att2_name att2_type att2_cardinality
           …
           attN_name attN_type attN_cardinality>
```

There are a few types of element content and when we declare this content we limiting structure of a data in document. Basic content models are (Dick, 2002):

- char data – <!ELEMENT Invoice_number (#PCDATA)>,
- elements – <!ELEMENT Invoice (Invoice_info, Supplier)>,
- mixed content – <!ELEMENT Description (#PCDATA | term)*>,
- empty content – <!ELEMENT E EMPTY>.

In examples mark | denotes the selection from several elements or content models. Comma (,) represents a sequence of elements. Sometimes, the occurrence of elements in XML document is multiple or optional, in DTD must be this recorded by marks following given elements or content models

- ? – zero or one occurrence;
- * – zero or more occurrences;
- + – at least one occurrence.

Without these occurrence marks each elements must be used just once. More information about XML and definition of various XML document can be found at XML specification (W3C – World Wide Web Consortium; 2010b).

Document modularization

In the XML world, the modularization is a decomposition of the complete set of elements and attributes (XML application) into at least two logical modules according to meaning and function of these elements and attributes. For example in the case of XHTML according to W3C the modularization refers to the task of specifying well-defined sets of XHTML elements that can be combined and extended by document authors, document type architects, other XML standard specifications, and application and product designers to make it economically feasible for content developers to deliver content on a greater number and diversity of platforms (W3C – World Wide Web Consortium; 2010a).

Definition: A standalone subset of elements and attributes of complete XML application be called *module* if this subset can be used individually as a standalone XML application and can be modified or extended, but still must be possible to use it as a part of another XML application.

After modularization one XML application consists of a few modules (see definition) that can be used according to needs. In other words, XML modularization enables flexible modification of the complex structures and their redefinition by using new modules or disabling existing modules. Applying on the random type of enterprise documents, it is necessary to specify individual modules according to their logical meaning and meaning of their parts.

Example figure (fig. 2) shows possible modularization of the basic enterprise document – invoice. All delimited parts of the invoice can be described and formalized in the separated way. The result is a collection of standalone parts with possibility to be used not only in this one document, but also in various ones.

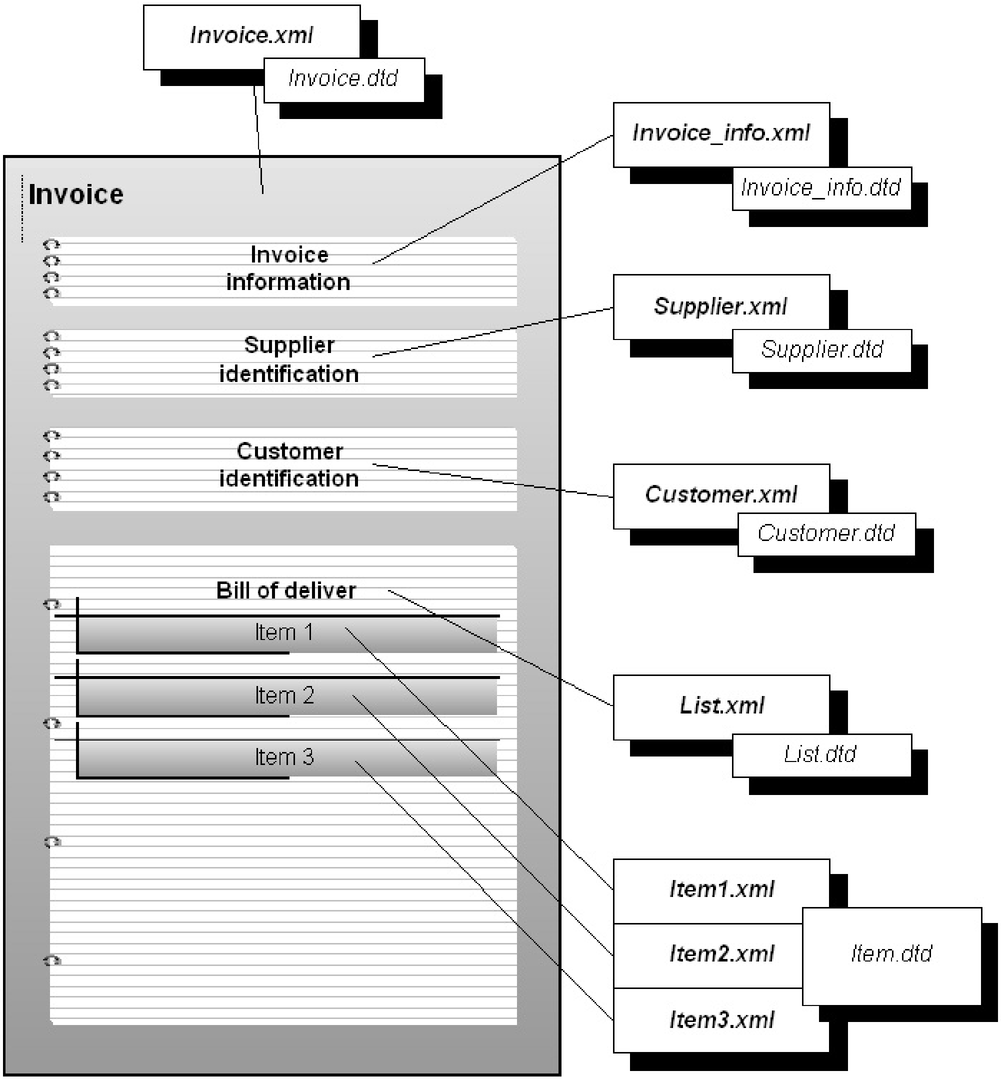
RESULTS

Reusing XML based documents

When considering XML as a basic for document reusability there are a few problems to be solved:

1. What types of documents can be reused?
2. How to determine fragments of any type of document to be reusable?
3. How to specify relations between fragments?
4. How to process fragments and work with documents?

In the field of XML documents mainly all types of documents can be reusable. Each partial document can be integrated into another complex docu-



2: *Examples of modules within invoice document*

ment by special techniques. However, techniques are static and they request to implicit record of relations between documents for example by entity references or XML inclusion tags.

In fact, each element from XML document can be reused. Although this granularity is usually too low, elements are the minimal parts of all XML documents that can stay separately.

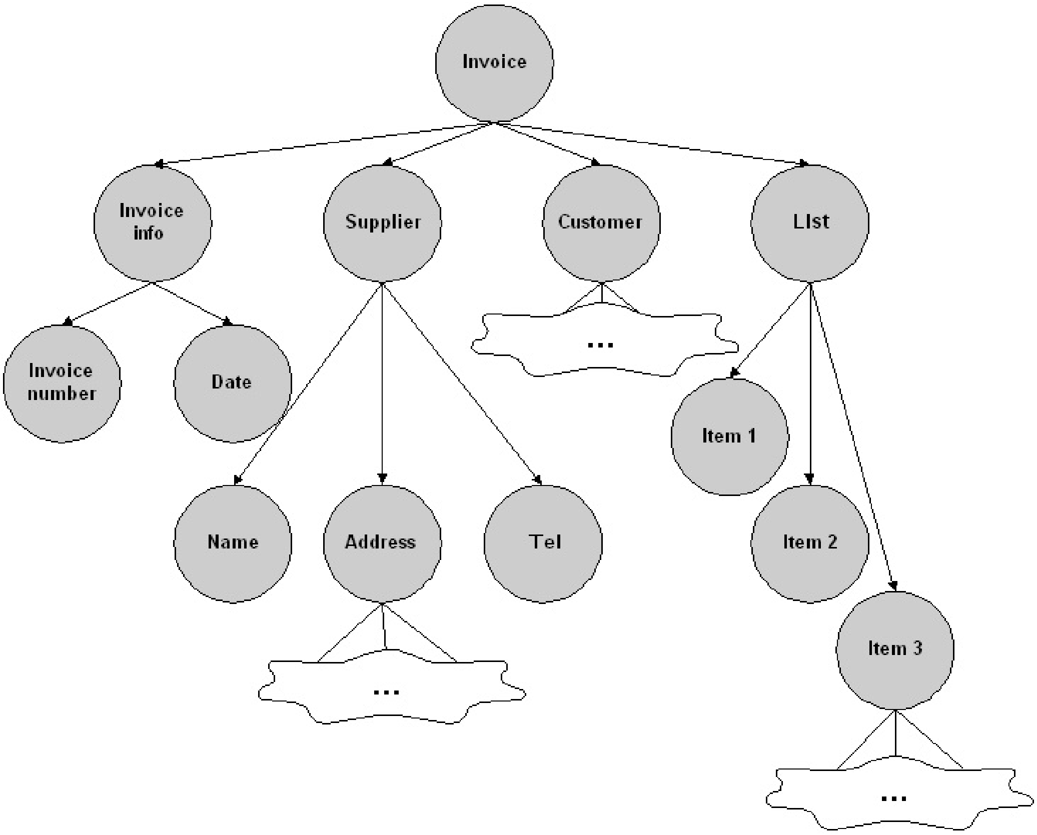
Fundamentally, any XML document (for example fig. 3) is a directed tree T = (N, E), where N is a set of nodes (elements) and E is an edge (relation) – path between nodes. Just one of nodes $n \in N$ is a root element, in which case all edges have orientation away from this root. Any node in a tree T, together with all the nodes below it, comprises subtree of T. But not every subtree is suitable to be delimited as a module. It is necessary to compare logical meaning of root element for this subtree with business logic. If the XML represented by this subtree described standalone entity then elements should be modularized.

The algorithm for delimiting modules for one type of XML document consists of next steps:

1. Formalize DTD (or another schema) for given type of XML document. Be X this schema. Construct a tree T_x based upon formalized schema.
2. Find all possible subtrees of T_x.
3. For each subtree of T_x check if XML represented by given tree could be used as a standalone entity. If yes, sign subtree.
4. Set modules for all signed subtrees.

If modules are defined then basic XML documents can be prepared for example by template-based system. But for preparing complex documents that compose more basic documents all necessary relations must be defined.

Enterprise documents are usually processed within a system called Document Management System of Content Management System. XML document can be saved by means of special data types of XML native databases. But the way of their collating



3: *Example of XML tree*

is not essential in comparison with their processing. During processing XML fragments and document should be accepted following rules:

1. Each XML fragment is saved autonomously; there are not duplicate fragments within base.
2. After processing the result document must be well-formed XML document treatable with standard XML tools.

Basic techniques for processing modularized XML

There are two basic approaches how to use modularized XML documents. Both are being used today within XML processing.

Entity references

An entities and entity references are standard parts of the XML specification. An XML document may consists of one or many storage units called entities with own content and name. In our case each of XML files can be considered as an independent entity. Joining them together required a special mechanism called entity reference which refers to the content of a named entity. For our purposes we will discuss only parametrical external parsed entities but there are more other in the XML specification. A parametrical external parsed entity can be described as a XML fragment well-formed or non-well-formed.

Each of these entities must be declared in document type definition. Syntax is trivial.

<!ENTITY entity_name SYSTEM “xml_fragment.xml”>

In our example case (fig. 2), when creating Invoice.xml, entities should be declared in Invoice.dtd for example as follows

<!ENTITY invoice_info SYSTEM “Invoice_info.xml”>
<!ENTITY supplier SYSTEM “Supplier.xml”>
<!ENTITY customer SYSTEM “Customer.xml”>
<!ENTITY list SYSTEM “List.xml”>

The result document Invoice.xml itself has to refer these entities by using entity references:

&invoice_info;
&supplier;
&customer;
&list;

The same situation, but at the lower level, occurs for fragments with names ItemN.xml where N is number. Declaration for these entities should be found in List.dtd.

When processing XML document with entity references, every reference is considered as a link to another document and its occurrence is replaced by its content. The parser works with modularized content logically as with one complex document.

Although DTD is only one tool that can be used for formalizing XML applications, the demonstrated principle is similar for other language as XML schema.

Document inclusion

One of the great problems of using entity reference as the way how to modularize XML document is a necessity to prepare and process file containing declarations for all entities. It means DTD file. This fact is a barrier for processing XML which validity is based upon another formalizing language as XML Schema of NG Relax. Adding DTD useless is in these cases. That is why XML inclusion has been set.

XML Inclusion (XInclude)[2] specifies a special set of elements in own namespace that represents the place for inclusion of defined content when processing. The basic syntax is simple:

```
<xi:include href="Invoice_info.xml"/>
<xi:include href="Supplier.xml"/>
<xi:include href="Customer.xml"/>
<xi:include href="List.xml"/>
```

Using XInclude is a powerful technique how to join standalone XML fragments into the complex document. The specification describes not only simple inclusion as example shows but there are also possibilities to include only parts of given documents, handle error states or specify source content that should be parsed or unparsed. No declaration in DTD is needed. On the other hand, when processing XML with XML inclusion a special inclusion processor must be enabled. And this could be sometimes problem, not all parser are able to handle XML Inclusion tags.

CONCLUSIONS

Basic principles and techniques for reusing XML documents were initiated in the text. In fact, basic techniques are native for processing XML documents, and enterprises and enterprises' documents are only one of the possible domains. On the other hand, a lot of enterprises' documents are exemplary by given structure and templates, and that is why this is suitable domain for reusing partial components.

It is clear that a set of suitable types of documents involves especially documents created automatically for example within document management systems.

Next work in this area will be focused to preparing methodical approach for delimiting modules in various types of XML application and designing the system enabling simple using of reusable documents' parts.

SOUHRN

Principy znovupoužitelnosti podnikových dokumentů založených na XML

Problematika zpracování XML dokumentů se úzce týká uplatnění informačních a komunikačních technologií v podnikovém prostředí, neboť jazyk XML již delší dobu představuje platformu, na jejíž bázi je řešena řada softwarových problémů.

Jedním z nich je i problém tvorby a zpracování dokumentů, a to především v podobě strukturovaných a semistrukturovaných dat. Řada dokumentů je tvořena z dílčích komponent, které mohou ve své podstatě představovat samostatné dokumenty a v některých případech se jedná o komponenty nejen se shodnou strukturou, ale i obsahem. Jsou-li jednotlivé XML dokumenty uchovávány jako celky v rámci specializovaných XML databázích či alespoň datových typech, je tento fakt doprovázen i duplikováním některých částí dokumentů.

V této práci je poukázáno na principy znovupoužitelnosti a zpracování XML dokumentů na bázi jejich modularizace. Výsledky by mohly být aplikovány v rámci softwarových nástrojů jako jsou document management nebo content management systémy.

XML, XML modularizace, podnikové dokumenty

Tato práce vznikla rovněž v souvislosti s řešením VZ MSM 6215648904/03/03/06.

REFERENCES

BOLL, S., KLAS, W., WESTERMANN, U., 2000: Multimedia Document Model. In: *Multimedia Tools and Applications*, Volume 11, Number 3. ISSN 1380-7501.

DICK, K., 2002: *XML: A Manager's Guide.* Addison Wesley Professional. ISBN 0-201-77006-7.

RAMALHO, J. C., ALMEIDA, J. J., HENRIQUES, P., 1998: Algebraic Specification of Document. In: *Theoretical Computer Science*. ISSN 0304-3975.

[2] http://www.w3.org/TR/xinclude/

W3C – WORLD WIDE WEB CONSORTIUM.: *Modularization of XHTML™ 1.0*
[online]. c2010a, [cit. 2010-05-28]. Available from < http://www.w3.org/TR/xhtml-modularization/>.

W3C – WORLD WIDE WEB CONSORTIUM.: *Extensible Markup Language (XML)* [online]. c2010b, [cit. 2010-05-28]. Available from <http://www.w3.org/XML/>.

YAO, Y., TRAPPEY, A. J. C., HO, P., 2003: XML-based ISO9000 electronic document management system. In: *Robotics and Computer Integrated Manufacturing*, Volume 19, Issue 4. ISSN 0736-5845.

Address

Ing. Roman Malo, Ph.D, Ústav informatiky, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: malo@mendelu.cz