OASIS LegalXML eContracts TC

Requirements process

Requirements for clause model

Name:	Requirements for clause model
Owners:	Peter Meyer, Elkera Pty Limited, pmeyer@elkera.com.au,
	Jason Harrop, SpeedLegal Pty Limited, jharrop@speedlegal.com
Draft no:	1
Draft date:	27/05/03
Contributors:	

1. Introduction

1.1 Purpose of this document

This document sets out requirements for a model for the XML markup of clauses in a contract document that can also be used for the markup of the content of most, if not all other legal & business documents.

1.2 Definitions

clause broadly refers to a distinct term or set of terms in a contract document together with headings and numbering desired by the parties. The concept is imprecise and is explored by the examples listed in Attachment 1.

contract document means a document that is adopted by the parties as the record of the contract.

document means a hard copy or electronic representation of a written work.

legal & business documents means documents commonly created in law firms, government agencies, and business enterprises, including:

(a) contract documents;

(b) litigation pleadings;

(c) company constitutions;

(d) Deeds (Trust deeds, conveyances etc);

(e) minutes of meetings;

(f) articles, reports, advices and opinions;

(g) correspondence and memoranda;

(h) tender documents and proposals;

(i) training, quality and office procedure manuals;

(j) simple marketing and product information documents.

1.3 Related documents

The following scenarios have been presented to the TC:

- *Enterprise Contract Management*
- *Contract Generation Systems*
- *Click-Through Contracts for Software Downloads*
- *Automated Contract Negotiation*
- *Law Firm Contract Creation and Negotiation*
- *Dispute Resolution and Litigation Involving Contracts*
- *XML in enterprise documentation systems*
- *Data Consortium Contract Schema requirements*

2. Scope of these requirements

2.1 TC charter

The mission of the TC is to:

"... develop open XML standards for the markup of contract documents to enable the efficient creation, maintenance, management, exchange and publication of contract documents and contract terms."

2.2 Perspective on eContracts

Before a written contract can come into existence, draft contract documents must be prepared. The draft document may be prepared specifically for the particular contract or it may be a standard form that is adopted by the parties without negotiation on the wording of specific provisions.

The contract document or documents that represent a contract may be published in tangible (print) or intangible (electronic) form. For the purposes of these requirements the published form of the contract document is considered to be not relevant. It is for the parties to determine the mode of representation of the contract document.

Authors in most law firms and other enterprises who prepare contract documents also prepare many other legal & business documents as part of their daily work.

From a document preparation perspective, contract documents are not sui generis. Contracts may take the form of correspondence. They may consist of a few simple paragraphs or a complex set of numbered structures. They may be completely self contained with a description of the parties and signatures or they may be incorporated into a contract by other documents, orally or by conduct of the parties.

Complex text structures commonly used for contract terms are also commonly found in many other legal & business documents such as trust deeds, corporate constitutions, litigation pleadings, tender requests and technical documents.

2.3 The clause model focus

Different kinds of legal & business documents will have particular high level structures and metadata requirements. For example, contract documents may require provision for parties, background (recitals), body provisions (clauses) and schedules or attachments. A letter that contains contract terms may have a different overall structure and a corporate constitution may have yet another high level structure (eg, it does not have parties).

However, within each document the terms or provisions of the document can have an identical structure. This document defines the requirements for the markup of structures of the kind described in Attachment 1 that can appear in a wide range of legal & business documents.

2.4 Relationship to other TC work product

The eContracts Technical Committee is distinguishing between the "content" of a contract document on the one hand, and on the other hand, other information and practice which is relevant but not present in the contract document itself (called "transaction information").

This document is part of the work on the "content" of the contract document.

In its work on the "content" of a contract document, the TC is making a further distinction, between the "structure" of a contract document and its "semantic" content.

This document is part of the work on the structure of a contract document. Once these requirements are finalised, a clause model which meets these requirements will be devised. Once the TC has a basic clause model, other aspects of the structure will be addressed (for example, signature/execution blocks, cross references etc). For further details, please refer to section 8 (Development Approach).

Clearly it will be necessary to relate the semantic content identified in the clauses to the structural model of those clauses. The way in which that is done is outside the scope of the clause model process, and will be tackled later in the semantic modelling exercise.

2.5 Scope of this clause model

Clause structures of contract documents and other legal & business documents may include:

(a) quotations, examples and amendments which are often distinguished from the rest of the text by indention, font changes etc in published documents;

(b) tables;

(c) images;

(d) equations (mathematical formulae);

(e) cross references to other clauses or provisions of the document;

(f) citations to other works;

(g) defined terms;

(h) references to a party;

Requirements for these features are not fully explored in these requirements. It is proposed (see section 8 – Development Approach) that these requirements will be addressed by the TC once a clause model meeting the requirements in this document has been devised and adopted by the TC. In the meantime, the clause model to be developed may include stub or placeholder elements for the proposed objects to signify the overall way in which the model will work.

3. Business problems

3.1 Overall benefits of using XML for document markup

Enterprises involved in the creation of large numbers of contract documents and other legal & business documents may obtain many benefits from the use of XML markup for those documents, including:

(a) They will avoid the technological obsolescence of word processing documents that are tied to particular software products and even versions of those products.

(b) They can more reliably automate many of the processes involved in document creation in knowledge based systems.

(c) They can reliably and automatically produce documents in multiple publishing formats to meet hard copy and electronic publishing requirements.

3.2 The need for common tools across document types

The benefits from the use of XML markup for the preparation of contract documents are equally applicable to the preparation of many, if not all other legal & business documents.

Enterprises involved in the preparation of contract documents need to provide knowledge based systems, software tools, training and technical support to their authors. Wherever practicable, they will wish to use common tools for authoring, manipulating and publishing contract terms and contract documents to standardise procedures throughout the enterprise.

3.3 User training and change management

Enterprises that wish to introduce XML based applications for document creation, manipulation and publishing face substantial change management issues. In particular, authors of contract and other documents must be introduced to a range of new concepts to those applicable in current WYSIWYG (what you see is what you get) word processor based systems.

The use of divergent XML DTDs or schema will make it difficult to train and support authors of contract terms who may also create other legal & business documents using XML markup.

3.4 Implementation and support costs

The use of divergent XML DTDs or schema within an enterprise will make it costly for enterprises to implement software systems for authoring, content manipulation and publishing. Each DTD that performs essentially the same underlying function will involve a complete duplication of cost to the enterprise.

Likewise, developers of XML based software tools incur substantial costs to develop systems around particular XML DTDs or schema. Currently, most developers use proprietary DTDs or schema for their applications.

3.5 Proprietary DTDs and tied software

An enterprise that uses a DTD and set of applications provided by one tools vendor cannot easily adopt the software tools provided by another vendor. Usually, it is necessary to undertake extensive modification of the software or to convert the XML data to conform to the new DTD or schema. Existing systems are rendered obsolete and personnel must be re-trained. This negates many of the benefits of using XML for document markup.

3.6 Exchange of XML data

Just as today enterprises routinely exchange word processing versions of their draft contracts and contract documents, they will wish to do the same with XML format documents if XML is widely adopted.

There are a range of problems that need to be addressed to facilitate exchange of XML data at even the most basic level. These include:

(a) Do the parties need to be able to create the same visual representation or display of the XML document?

(b) If so, how should this be accomplished, bearing in mind that different enterprises may use software provided by different vendors?

(c) If the recipient of an XML document wishes to edit the document, how will he or she apply the same numbering scheme applied by the author? In particular, how will he or she be able to apply the same automatic numbering to document objects as was applied by the author using a different vendor's software?

(d) How will the recipient interpret internal cross references in the document? Again, how will he or she be able to update or edit these cross references in a document created by someone else using a different vendor's software?

(e) Should users of the standard be able to apply metadata to the document or to objects in the document to suit their particular information management and processing needs? If so, how will recipients of XML documents deal with enterprise specific metadata in the markup they receive?

(f) Is the standard expressed as a DTD, W3C Schema, and/or Relax NG? If more than one, which is definitive?

(g) Should users of the standard be able to augment the DTD by adding additional element and attribute declarations? If so, how will this be handled by recipient's software?

None of these issues directly affects requirements for the clause model at the level approached in these requirements. Some issues will be relevant to the final clause model but all must be addressed before the overall standard can be completed.

3.7 The need for widespread use of XML markup

Unless XML markup is used widely by law firms and enterprises to create contract documents, it will not be practicable for them to transmit to or exchange XML format documents with counterpart firms or agencies such as regulatory bodies and courts.

If only one party to a transaction is using XML markup for documents, the XML markup must be shielded from the other parties by the document publishing process. The only benefits to be obtained from the use of XML markup will be those that flow from more efficient document production processes within the creating enterprise.

To encourage widespread adoption of XML markup to create contract documents, that markup should be readily applicable to other legal & business documents.

The conclusion from these points is that law firms and other enterprises will not adopt XML markup for contract documents on a large scale unless:

(a) they are able to use DTDs or schema that at least provide for common markup models for contract documents and other legal & business documents;

(b) the DTDs or schema are simple to use for authors with a wide range of skills;

(a) the DTDs or schema are supported by multiple software vendors; and

(b) the DTDs or schema are substantially the same as those used by counterpart firms and enterprises.

If a formal standard DTD or schema is not developed, these needs can be met only by the adoption of a proprietary DTD that slowly evolves into a de facto standard.

3.8 The Court Document 1.1 DTD

The OASIS LegalXML Court Document TC has produced the XML Court Document 1.1 Proposed Specification (http://www.oasis-open.org/committees/legalxml-courtfiling/documents/court_document/courtdocument_11(rev1).html). Section 5.7 of the Proposed Specification states:

> “2 The Court Document 1.1 XML specification should provide XML markup to describe the structure of legal court documents, typically pleadings, orders, briefs, discovery documents, and the like filed in lawsuits or formal administrative hearings or associated with lawsuits or hearings and having the following required, but not exclusive, characteristics:
>
> …
>
> - the organization of the contents of the body of the court document into basic structural components of paragraph groups and paragraphs and XML elements generally describing their sub-components, including lists, phrases, quotations, footnotes, and multimedia objects;

> …”

Section 5.2 of the Proposed Specification states:

> “The Court Document 1.1 DTD marks a beginning by describing the organization and structure of the information contained in court documents. It includes some general XML tags for marking up phrases, quotations, lists, and similar structural items that often appear in the body of court documents. Through the use of attributes, it also allows for semantically meaningful descriptions of such structural elements. This specification assumes, however, that more specialized standard XML vocabularies used in particular areas of legal practice will have to be developed over time and that a means should be provided to allow authors to mark up specialized legal terms within XML court documents. The specification provides a basic framework and general starting point from which further development can proceed.”

Further, section 6.4.6 of the Proposed Specification states:

> “The Court Document 1.1 DTD organizes the text contents of the body of an XML court document into a structural hierarchy of paragraph groups (i.e. sections), paragraph subgroups (i.e. subsections), paragraphs, and subparagraphs. Paragraph subgroups and subparagraphs extend to three levels, thus establishing a hierarchical structure that includes sub-sub-subsections and sub-sub-subparagraphs. …
>
> Within paragraphs, the basic container of text in the body of an XML court document, the Court Document 1.1 DTD allows the use of various elements to distinguish footnotes, lists, citations, phrases, quotations, and similar items of text content. ….”

It is clear from these statements that the Court Document TC has not set out to deal with the broader strategic issues of the use of XML for the markup of legal & business documents in law firms and other enterprises considered in these requirements. The Court Document DTD does not set out to provide a generic model for the markup of content found in those documents. The Court Document DTD does define a structural hierarchy for the content of litigation documents, an approach consistent with that proposed in these requirements.

The key issue in development of another standard in this TC is that the costs and technical issues involved in implementing XML based systems make it highly unlikely that law firms and other enterprises will adopt one markup model for pleadings, orders, briefs etc, another for contract documents and yet another for correspondence, advices and other legal documents.

During development of the clause model it will be necessary to analyse the Court Document DTD to determine the contribution that might be made by utilising approaches taken in that DTD.

4. Vision statement

The objective of these requirements is to develop a standard for the markup of contract documents that will provide low cost, convenient access to the benefits of using XML to the widest range of law firms and other business enterprises. To achieve this objective it is necessary to:

(a) provide a common model for XML markup of all legal & business documents by law firms and other enterprises;

(b) develop a standard that will encourage the development of competitive applications that can be used by enterprises to more effectively prepare, manage and publish contract documents;

(c) develop a standard that will minimise the cost and risk of implementing XML systems for the preparation of contract documents and other legal & business documents;

(d) develop a standard that will enable law firms, enterprises and agencies who wish to exchange documents in XML format to do so effectively using XML based software applications of their choice; and

(e) avoid dependencies between enterprise data in XML format and software systems of particular vendors.

5. Analysis of a clause

5.1 The clause structure example in Attachment 1

The structure example shows a series of numbered or bulleted components that represent a hierarchy of concepts. When displayed in visual form, this hierarchy can be discerned by a reader from clues provided by numbering, indentation, font size or formatting. If these clues are removed, the text may become unintelligible.

All of the hierarchical relationships shown in the structure example can be found in a single contract document. All but the simplest contract documents will contain at least a subset of those relationships.

5.2 What is a clause?

In different enterprises and in different jurisdictions different names will be applied to particular objects. The terms “article”, “clause”, “section” or “paragraph” might all refer to the same structural object.

Lets consider which of the numbered objects in the example is a clause:

- The object numbered “1.”,
- The object numbered “2.”,
- The object numbered 1.2,
- The object numbered 1.3,
- The object numbered 1.3.1,
- The object numbered 1.4.1?

The object numbered “2.” has paragraph text inside it while the object numbered “1” has only other headings. Is one a clause and the other not?

Similarly, with the objects numbered “1.2” and “1.4”.

The example shows that objects at the same level in the hierarchy may contain paragraph text or another structure similar to itself.

Which one of the objects numbered “1.3.1” and that numbered “1.4.1” is a subclause? Even though both objects are at the same level of the hierarchy, only one has a heading. Does this change its designation?

Which of the objects should appear in a contents listing to the document? How will the author signify his or her intention or is it something beyond his or her control? How will the application developer determine which objects should be listed in the table of contents?

If a numbered object similar to object “2” appears in contract, a litigation pleading or a legal opinion will it always be given the same form of designation or citation? Might it be referred to as a “clause” in one context and a “paragraph” in another?

5.3 What is a list?

Most would probably regard the objects numbered 1.1(a) to (g) as a list. Some might regard the objects numbered 1.3.1 to 1.3.3 as a list while others might refer to them as subclauses and others as clauses. Some might change their mind according to the numbering scheme used.

Does the presence of introductory text, such as at the start of object 1.1 affect the designation of the items that follow?

Should it be necessary for authors to understand and consistently apply these distinctions when creating XML markup?

Which are the logical boundaries between objects in the example? In object “1.1”, where does the object numbered “(e)” conclude?

5.4 The use of numbering

In contract documents it is common, but not essential, that terms are numbered.

If the decimal numbers were removed from the headings in the example would this change the designation of the structures?

Whether numbers are included with clause objects is a matter of convention and choice by the drafter. Once included, it is assumed that they are a significant part of the structure.

5.5 Conclusions and assumptions

Any of the objects in the decimal numbered hierarchy might be called a clause. This term and others such as "article", "section" or "paragraph" have no fixed meaning.

There is no clear distinction between a clause, subclause, list item or list paragraph. Any of these terms might be applied to object 1.3.1 in the example.

Authors should not have to consistently apply markup that makes subtle distinctions as to whether objects are clauses, subclauses or lists since these terms have no clearly defined meaning in structural markup.

It will be important to capture the hierarchical relationship of objects in the XML markup of a clause so that the appropriate visual clues to that relationship and the meaning of the clause can be conveyed in the rendered output to readers of the document.

6. Approach taken to needs identified in the scenarios

The scenarios identify many needs that may or may not require a particular feature in clause markup. For example, in the *Law Firm Contract Creation and Negotiation* Scenario it is stated that it should be possible:

“

- to see the status of a clause (agreed, acceptable, not acceptable, amendments proposed etc)
- to identify the changes between different versions of a clause;
- to capture the reasons for a change …”

Some users of XML markup for contracts may require some of these properties reflected in markup while others will not. Whether they will do so may depend on the approach taken by particular software applications. Information might be stored separately from the actual markup using database systems.

The common, high level requirement that can be extracted from the scenarios is that the XML markup should define clause objects so they can be uniquely addressed and processed by software systems.

It is proposed that the clause model should avoid addressing more specific needs in the first instance. The initial clause model will contain only the simplest markup necessary to represent the ordered, structural hierarchy of the document – as described in these requirements.

If a model meeting these requirements is achieved, it may be appropriate to examine more specific needs in more detail to determine whether they are sufficiently general to be part of a standard or whether they may add a burden for the many, to the benefit of only a few stakeholders.

The assumption is that the proposed structural, hierarchical representation will act as a common skeleton on which additional features can be added to meet the more specialised needs of particular interest groups.

7. Specific requirements

1. The clause model adopted for the markup of contract terms must be able to markup the core structures found in contract and other legal & business documents similar to those described in Attachment 1.
2. The clause model markup must represent the structured hierarchy of the content so that it can be rendered in a form to accurately convey the structure and meaning of the text to readers. In the example in Attachment 1, object 1.1, the markup must ensure that the words “but excluding violet,” are part of item (e). The words “from which all colours can be derived.” are part of the introductory grammatical paragraph so that one element container must include everything beginning with “Here is ..” to “derived”.
3. The clause model must be able to represent the “benchmark” contracts which are submitted by members of the TC and accepted by the TC as in scope.
4. The clause model must define clause objects so they can be addressed and manipulated by software systems as self contained objects.

5. The clause model must provide for self contained markup of content so that if the parties desire to use the XML text file as the contract document, it is not necessary to use any software apart from that needed to display a text file to determine the terms of the contract. For example, other software should not be required to interpret numbering of objects or cross reference targets.

Note:

It is hard to envisage a situation where the parties would wish to use an XML file as a contract document. Normally, the parties will translate the XML document into a display or print format such as HTML, PDF, RTF or any other format convenient to the parties. The objectives of this requirement are twofold:

- to ensure that XML documents can be exchanged between parties in a way that does not impose requirements to use particular software to effectively interpret the document; and
- to ensure that if the XML file is used as the contract document the parties do not have to use proprietary software to prove the terms of their contract, unless they have specifically chosen to do so.

6. The terminology or element names must not imply any particular form of citation, so that the clause model can be used for different types of document by persons from jurisdictions with different citation traditions. Consequently, the DTD/Schema must not use the following terms in element markup:

(a) article

(b) clause

(c) recital

(d) rule

(e) section

(f) paragraph

(g) chapter

(h) part

(i) division

(j) regulation

(k) schedule

(l) attachment

(m) annexure

(n) exhibit

(o) appendix.

7. The clause model must permit the markup of contract terms without inclusion of any legal semantic markup or annotation. In other words, the XML markup must not be relevant to the interpretation of the document once it is rendered in a published form unless the parties specifically agree otherwise.

8. The clause model must be as simple as practicable to facilitate user training, support and application development. To make it easy for users, semantic distinctions between generic objects will only be introduced where the benefits are clearly demonstrable.

9. The model must allow document authors to re-use content in different levels of the hierarchy, without having to change the names of the elements in order for the document to remain valid.

10. The model must allow clauses or other content to be incorporated into a document by reference.

11. Once specific requirements for these features are determined, the clause model must provide for:

    (a) the numbering of objects and the means by which numbers are specified in the markup;

    (b) inclusion of distinct content, namely:

        (i) quotations,

        (ii) examples and explanatory notes,

        (iii) amendments to the terms of other contracts,

        that are part of the contract document;

    (c) inclusion of tables with table cells that may contain other clause model structures;

    (d) images;

    (e) equations (mathematical formulae);

    (f) footnotes / end notes;

    (g) internal cross references;

    (h) citations to other works;

    (i) defined terms;

    (j) references to parties;.

8. Development approach

It is proposed that development of a clause model to satisfy these requirements will proceed in stages so that the issues to be considered at each stage are limited. The proposed stages are:

(a) Develop “basic” clause model (ie excluding objects identified in requirement number 11, other than stub or placeholder elements).

(b) Once the basic clause model is agreed, develop requirements for issues set out in requirement number 11.

(c) Develop “complete” clause model (ie covering the requirements developed for the issues set out in requirement number 11)

(d) Develop requirements for markup of complete contract documents using the clause model.

Attachment 1 – Clause structure example

The following example is included to identify the range of structures intended to be covered by the clause model.

1. Provisions about the specification of colours in contracts

1.1 Spectrum colours

Here is a contrived, complex list structure using the spectrum colours and one or two others:

(a) red,

(b) orange,

(c) yellow,

(d) green,

(e) blue, including:

> (i) pale blue,
>
> (ii) dark blue,
>
> but excluding violet,

(f) indigo, and

(g) violet,

from which all colours can be derived.

1.2 CMYK colours

CMYK colours (cyan, magenta, yellow and black) are normally specified for inputs to colour printing processes.

1.3 RGB colours

1.3.1 RGB colour (red, green, brown) specifications are used for computer screen displays.

1.3.2 Using only these 3 colours, you can specify any colour.

1.3.3 The number of colours you can specify depends on the colour depth available. For example:

> (a) 8 bit colour can render 256 colours
>
> (b) 16 bit colour can render 65,536 colours.

1.4 Using black and white

1.4.1 Greyscale

The number of greys depends on the available colour depth, as for other colours.

1.4.2 Black and white

This is really called monochrome. You can specify either:

- black, or
- white.

2. Colour profiles

One thing to remember is that when working with colours, always use a colour profile that is available for your display or output device. This will ensure you achieve the most consistent results.