

# Requirements for technical specification

## OASIS Legal XML eContracts TC

Document title: Requirements for technical specification  
Author: Peter Meyer, pmeyer@elkera.com.au  
Contributors: Jason Harrop, Zoran Milosovic, Rolly Chambers, Dr Leff  
Version no: Draft 0.02  
Date submitted: 21 December 2004

### Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Purpose of this document.....	3
1.2	Revision history.....	3
<b>2</b>	<b>Interpretation.....</b>	<b>3</b>
2.1	Definitions .....	3
2.2	Use of data flow diagrams.....	5
<b>3</b>	<b>Relevant contract transactions .....</b>	<b>5</b>
3.1	Sources of information .....	5
3.2	Types of contracts.....	5
3.3	Contracts with negotiated narrative terms .....	6
3.4	Ticket contracts.....	14
3.5	Standard form business and consumer contracts.....	15
3.6	Click through contracts.....	15
3.7	Electronic commerce contracts.....	17
3.8	Computer negotiated contracts .....	18
<b>4</b>	<b>The use of XML markup for contract documents.....</b>	<b>19</b>
4.1	The purpose of this section.....	19
4.2	The use of XML markup in contract authoring today .....	19
4.3	Available tools for authoring contract documents using XML .....	20
4.4	Current use of contract metadata .....	20
4.5	Current use of embedded data value markup for contracts.....	20

4.6	Machine generated XML markup.....	21
4.7	The need to transform XML markup into display formats.....	21
4.8	Exchange of XML contract documents between the parties .....	21
4.9	Likely future developments .....	22
4.10	Conclusions on the use of XML markup for contract documents .....	23
<b>5</b>	<b>The use of deontic contract languages .....</b>	<b>23</b>
5.1	Review of current use .....	23
5.2	Likely developments with deontic contract languages .....	24
5.3	Conclusions on the use of deontic contract languages .....	25
<b>6</b>	<b>Problems, objectives and scope .....</b>	<b>25</b>
6.1	Purpose of this section .....	25
6.2	Contract drafting and negotiation .....	25
6.3	Click through contacts .....	26
6.4	Contract management – transaction activities .....	27
6.5	Contract management – contract document.....	28
6.6	Electronic commerce contracts.....	28
6.7	Computer negotiated contracts .....	29
<b>7</b>	<b>Overview of the proposed specification .....</b>	<b>30</b>
<b>8</b>	<b>Functional requirements.....</b>	<b>31</b>
8.1	Approach to requirements development.....	31
8.2	Precedent contract documents .....	34
8.3	Contract document creation.....	36
8.4	Assent by parties to the contract.....	46
8.5	Contract management – contract activities.....	47
8.6	Contract management – contract document.....	48
8.7	Other processes.....	50

# 1 Introduction

## 1.1 Purpose of this document

The mission of the TC is to provide a standard for XML markup that should enable “the efficient creation, maintenance, management, exchange and publication of contract documents and contract terms”.

This mission statement provides a broad scope for the TC's work but does not define specific problems to be addressed or why the use of XML markup or a standard may be beneficial. If the TC is to be able to develop a specification that will be useful, it is necessary to identify the real problems that occur in relation to contract transactions and a practical specification that is based on an understanding of the way relevant transactions are carried out and the needs of relevant users.

This document identifies the business problems relating to the preparation and management of various kinds of contracts, the persons affected by those problems and the business needs of those persons to overcome those problems. Within that framework, it defines the functional characteristics an XML application must have to meet those needs.

Those requirements will enable the TC to design a technical specification to satisfy the identified requirements.

## 1.2 Revision history

Document versions are listed in the table in reverse chronological order.

Date/version	Description	Author
Draft 0.02 21/12/2004	Extensive revisions after TC comments on first draft, including contributions from Rolly Chambers, Dr Leff, Zoran Milosevic and Jason Harrop.	Peter Meyer
Draft 0.01 5/10/2004	Initial draft after San Francisco face to face meeting with background information, vision statement and outline of requirements topics.	Peter Meyer

# 2 Interpretation

## 2.1 Definitions

In this document:

***assent*** to a contract means the signification by a party that it wishes to be bound by the contract. Assent may be signified by a wide range of conduct or actions and is not limited to hand written or electronic signatures.

***authoritative contract document*** means the document or documents adopted by the parties as the authoritative record of the terms of their contract. This

document is also the document that the parties would use to prove the terms of their contract to an arbitral tribunal.

**contract** means an agreement between parties that is intended to be legally enforceable. A contract may be oral, partly oral and partly written or wholly recorded in writing. The terms of a contract may be contained in many contract documents.

**contract document** means a document that records some or all of the draft or agreed contract terms. Contract terms are traditionally expressed in a natural language but it is assumed that some or all the terms of a contract could be expressed in a deontic contract language. In these requirements, contract terms are assumed to be expressed in natural language unless otherwise stated.

**contract metadata** means information about a contract, particular contract terms or embedded data values that is not part of the contract narrative.

**contract narrative** or **narrative** refers to the terms of a contract expressed in natural language.

**deontic contract language** means a language that can express the rights and obligations of parties to a contract in a form that can be parsed by software applications and processed with other data to determine contract states.

**document** includes information in printed (hard copy) or electronic form.

**embedded data value** means a piece of information such as a product or service description, date, name, address, quantity or monetary amount that is embedded in the natural language expression of the contract terms.

**generic structural markup language** defines an ordered hierarchy of narrative components of a document such as clauses and paragraphs in a way that enables processing applications to determine that a marked up component belongs to a particular genus such as a clause, paragraph or list item. Such a language does not usually, on its own define anything about the (legal) function of a particular marked up object. For example, in a contract document, the generic structural markup would not differentiate between a payments clause or a warranties clause. Generic structural markup is commonly accompanied by metadata and embedded data value markup that may provide information about the function of particular objects.

**machine readable** information in a contract document refers to information about contract rights, obligations or states, that can be extracted from the document by a computer system. It does not refer to the reading of text unless the meaning of that text can be determined by the computer system. For example, a monetary amount that can be read from the text is not machine readable information unless the system can determine useful information about the statement of that amount in the contract such as who must pay it, to whom it must be paid, at what time is it to be paid or for what purpose is it paid.

**natural language** includes the mode of expression of contract narrative as it is commonly written by lawyers.

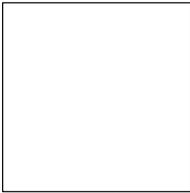
**precedent contract** means a document that is used by the drafter of a new contract document as a starting point or template to assist in creating that new contract document.

*TC* means the OASIS Legal XML eContracts Technical Committee.

## 2.2 Use of data flow diagrams

Data flow diagrams are used to identify the important processes, data flows and interfaces for each type of contract transaction considered in these requirements. They provide a generalised view of common situations for analytical purposes.

The symbols used in the data flow diagrams are explained in the following table.

	<p>Process</p> <p>An action or activity performed on data, either manually or automatically.</p>
	<p>Interface</p> <p>A person or system (document) that interacts with a process to either receive or input data.</p>
	<p>Data store</p> <p>A storage system for data used by the process.</p>
	<p>Data flow</p> <p>This shows the flow of information between an interface or data store and a process.</p> <p>A dotted line shows an optional flow.</p>

## 3 Relevant contract transactions

### 3.1 Sources of information

In 2003 various TC members contributed scenario documents describing contract transactions and their anticipated needs from the specification. These are available for inspection at ##.

In August and September 2004, some TC members prepared use case statements to define in more precise terms the actions undertaken by users in particular contract transactions. These use cases are available at ##.

The overall framework used in this document is based on the use case analysis undertaken at the face to face meeting of TC members in San Francisco on 19 and 20 September 2004.

### 3.2 Types of contracts

The following sections define the contract types identified as posing business problems that may be addressed by an XML technical specification.

The TC is not concerned with contracts that are wholly oral. Only those contracts that are at least partly evidenced in writing are considered. A written contract includes a contract where the terms are recorded only by electronic means.

### 3.3 Contracts with negotiated narrative terms

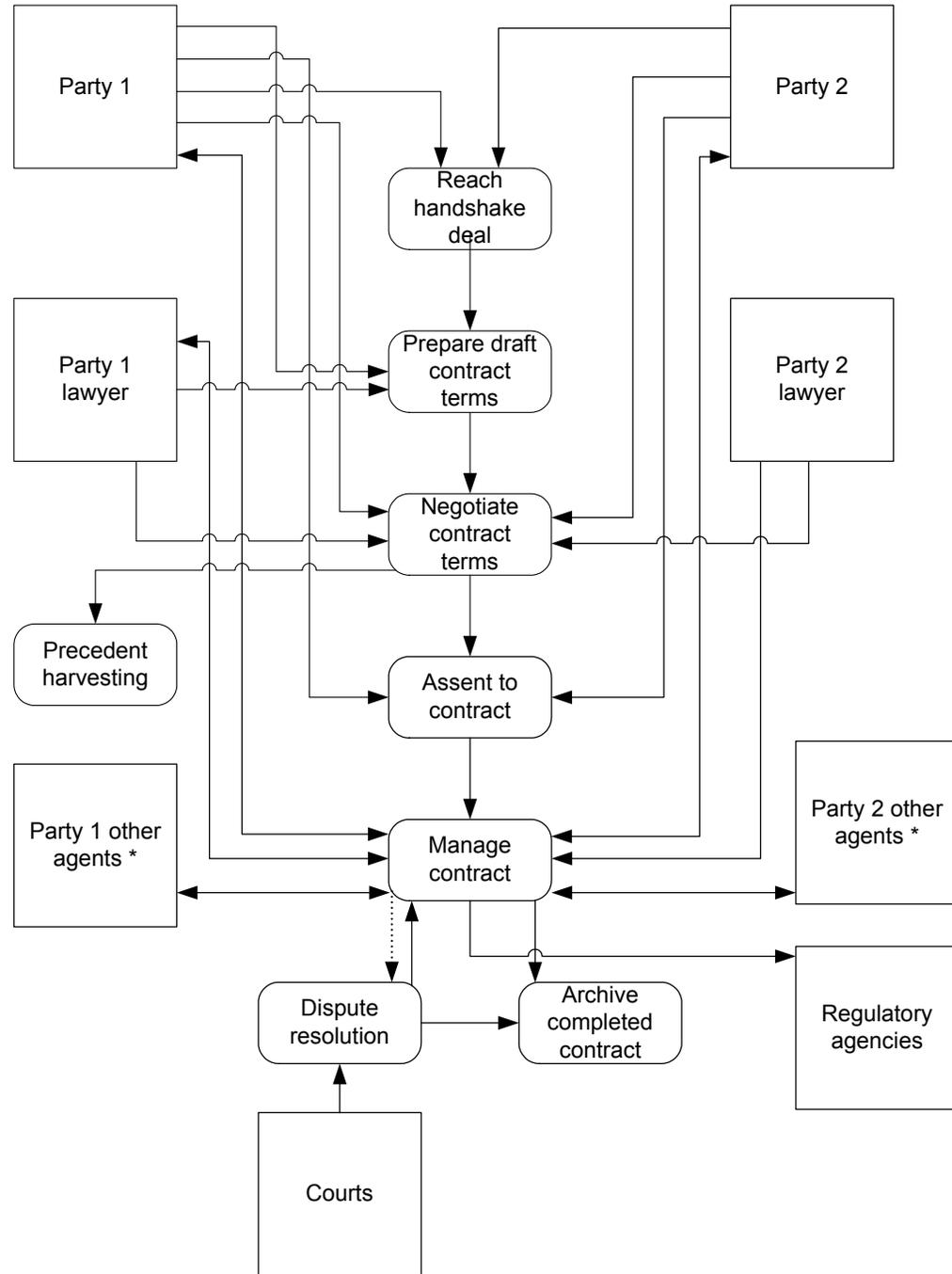
#### 3.3.1 Description

This case describes contracts that involve a negotiation over some or all the narrative terms of the contract.

Contracts with negotiated narrative terms are usually, but not exclusively, prepared by lawyers on behalf of their clients. These contracts are the most common contract documents created by law firms and corporate and government legal departments. They may include contracts with negotiated terms where a lawyer or legal dept. has drafted a precedent contract but a non-lawyer, such as a contracts manager, then negotiates the final narrative terms and prepares the final contract document.

The life cycle of contracts with negotiated terms is shown in Figure 1.

**Figure 1 Overview of negotiated contracts life cycle**



\* Other agents of the parties includes persons such as architects, accountants and other consultants etc. assisting with or interested in the transaction.

### 3.3.2 Handshake deal process

During the negotiation of a new transaction it is common for the parties to reach a business agreement or understanding that may or may not be a contract.

In commercial transaction it is common for the parties to prepare a term sheet or heads of agreement document that records the key features of the transaction. Such documents may or may not be legally binding contracts, depending on such factors as the expressed or implicit intention of the parties and the degree of certainty of the recorded terms. Even if a contract exists, the parties may proceed to replace the initial heads of agreement with a formal document that provides a more complete coverage of their agreement.

Term sheets or heads of agreement documents can be prepared using the same processes as contract documents. The handshake agreement process does not raise any distinct issues for these requirements.

### 3.3.3 Prepare draft contract terms process

The common components of this process are described in Figure 2.

Almost without exception today, the drafting of custom contract terms is carried out using word processing software such as Microsoft Word. For all practical purposes, no drafters, including lawyers are using XML based editing tools for this purpose. However, some software tools for document assembly may rely on precedent documents being stored in XML format to facilitate software processing. Contract drafters using such software normally would be unaware of the use of XML behind the scenes. This is referred to as *back end XML*.

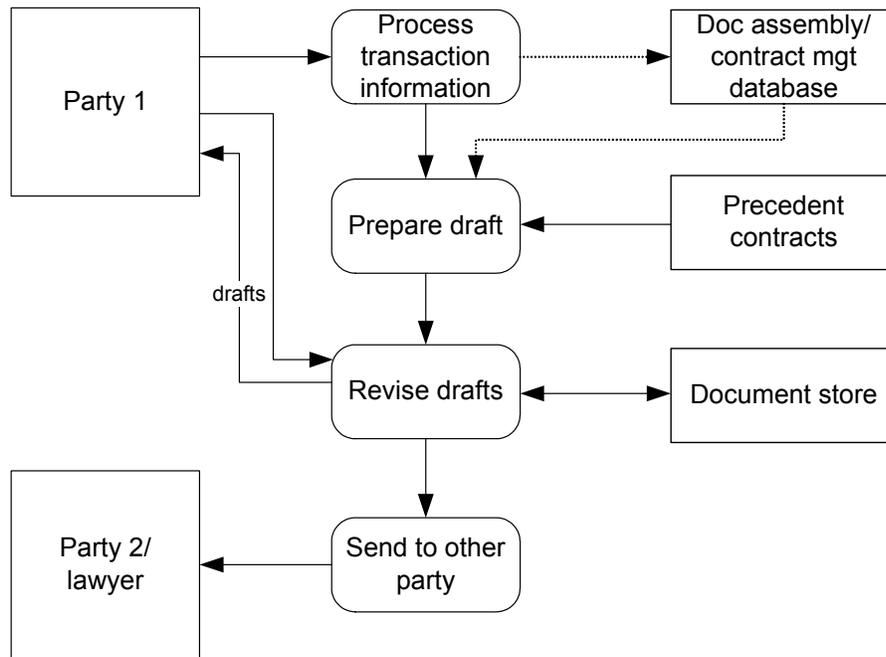
A possible exception involves contract management systems that provide are able to provide two way conversion between, say, XML and Word. This can be achieved where the authoring environment is tightly controlled to produce Word documents that strictly conform to a desired template.

Commonly, document assembly systems store transaction data values such as names, addresses, monetary amounts and dates. These data values may be automatically inserted into the draft contract document as embedded data values.

The draft contract documents so created may be based entirely on precedent contract terms, entirely on custom drafted terms or a mixture of precedent and custom terms.

When creating contract documents, authors commonly need to create other transaction documents such as letters, forms, notices and minutes of meetings. These documents may be created using the same precedent and document assembly systems as are used for draft contract documents.

Draft contract documents must be communicated to clients and other interested persons. It is often desired to provide access to the content of draft contract documents in various formats such as RTF, PDF, HTML etc according to the circumstances.

**Figure 2 Contract drafting processes**

### 3.3.4 Negotiate contract terms process

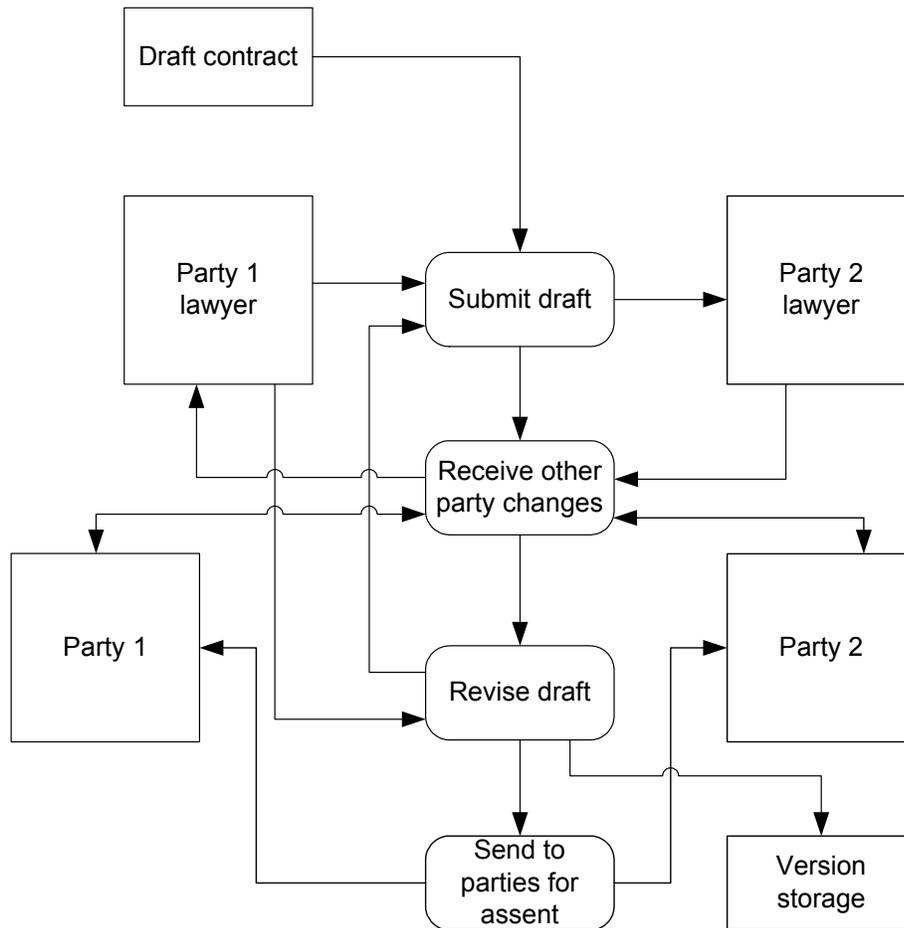
This process is described in detail in Figure 3.

A draft contract is submitted to the other party for approval. Where the terms are not highly standardised, the other party may propose changes to the draft. Often this will involve clarification of the actual commercial terms of the deal.

Changes proposed by the other party must be reviewed with the client by the contract drafter. Typically, some changes will be accepted, some rejected and new terms may be added.

The revised draft contract document is again submitted for review. This process continues until both parties are satisfied that the contract document reflects the deal. A formal contract document is then prepared for assent by the parties.

In some larger transactions the negotiating parties may want shared access to draft documents so they can contribute amendments, new terms and comments. Software collaboration platforms may be used to assist this process. This approach can be more convenient than the traditional approach where one party makes all changes to a draft document, sends it to the other party for review and awaits return of a package of comments and suggested amendments before repeating the process. This collaboration process involves making draft contract terms available in a variety of formats such as RTF, PDF and HTML, according to the role of the recipient.

**Figure 3 Negotiating the terms of a draft contract**

### 3.3.5 Assent to contract process

Assent to a contract with negotiated contract terms is usually signified by the parties signing a printed copy or copies of the contract document. It is relatively uncommon for parties to apply digital signatures to electronic contract documents under this scenario.

It is not uncommon for the parties to make last minute changes to contract terms at the time of assent. This may involve correction of a drafting error, an agreed change or the insertion of information that is dependant on the actual time of assent. Such changes may be noted by hand on the printed contract document if there are no facilities for editing the electronic source files and printing new documents.

The document bearing the signatures of the parties becomes the authoritative contract document.

### 3.3.6 Precedent harvesting process

Contract drafters are often concerned to capture new contract terms for later re-use in other, similar transactions. This is part of their knowledge management process.

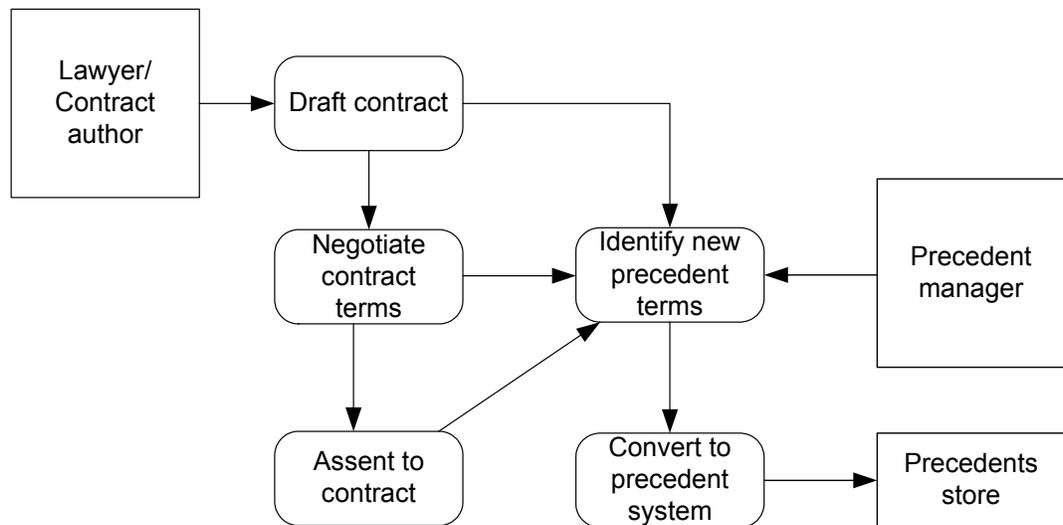
Precedent harvesting may be undertaken at various times in relation to the preparation of contract documents. For convenience it is shown in the overview in a slightly idealised form as occurring after the contract document terms are settled. The main components of the process are shown in Figure 4.

The precedent harvesting process does not apply only to contract documents. Lawyers apply the same process to other documents they prepare, including advices and litigation documents.

The harvesting process is quite difficult because of the bulk of content that must be reviewed manually. In practice it is difficult to require contract authors to categorise specific contract terms during the drafting process. Their attentions are directed to meeting the immediate needs of their clients and there is often little time to undertake additional work categorising contract terms.

If the enterprise uses back end XML for its precedent and document assembly systems, it will need to convert harvested content to that XML markup vocabulary. This conversion process may make it difficult to publish new precedent content quickly and it may add to the expense of maintaining the system.

**Figure 4 Harvesting precedent contract terms from transaction documents**



### 3.3.7 Manage contract process

#### 3.3.7.1 Description

Many, but not all, negotiated contracts are executory and involve ongoing obligations of the parties that may need to be monitored and managed until the transaction contemplated by the contract is completed or the comes to an end.

The contract management process is considered to involve two key sub processes:

- (a) manage the transaction activities (rights and obligations), as described in section 3.3.7.2.

- (b) manage the contract document and its publication to the parties, as described in section 3.3.7.3.

### 3.3.7.2 Contract transaction activity management

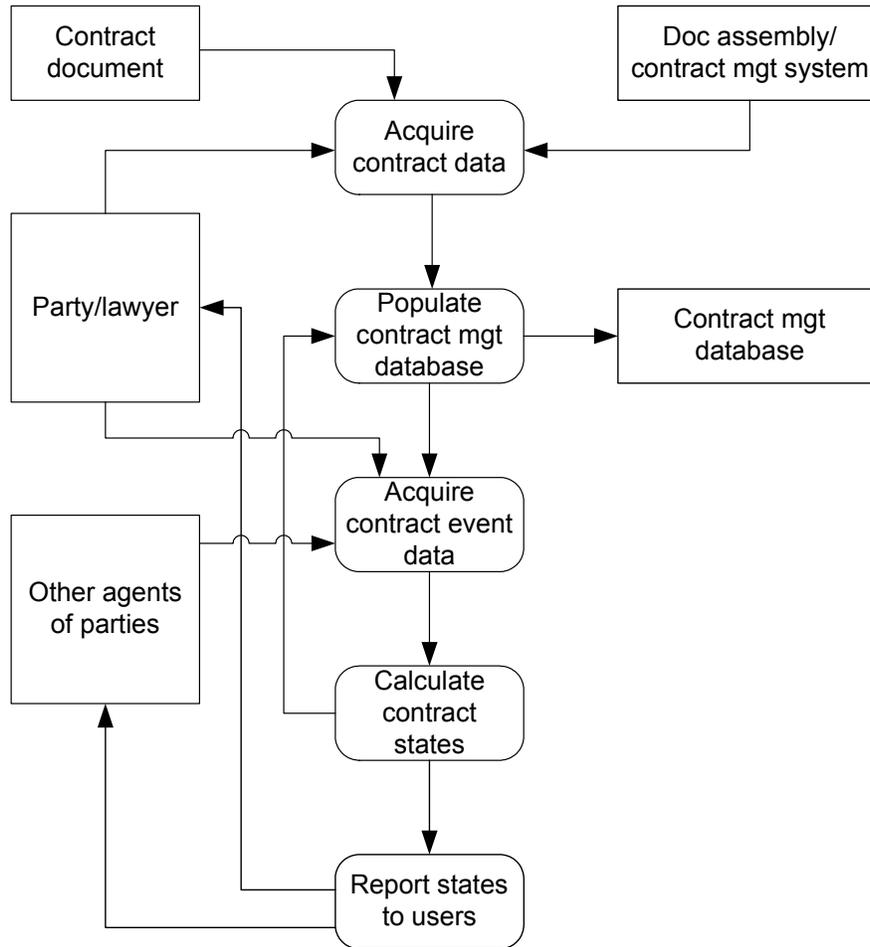
This process is described in Figure 5. It involves the management of the continuing contractual obligations of the parties.

Contract management can be undertaken at certain levels without software tools. At other levels, simple software tools such as spread sheets or desktop project management software (eg, Microsoft Project) may be used. In more complex situations, specialised contract management database applications are required to capture and manage the relevant information, calculate contract states and provide the reports desired by the interested persons.

The data needed for contract management may be derived from multiple sources:

- (a) The contract document is likely to be the source for only some of this information. At the very least, it will be common to input new information as contract events are undertaken and completed and as terms are varied.
- (b) Some information may already exist in a database system used to generate the contract document.

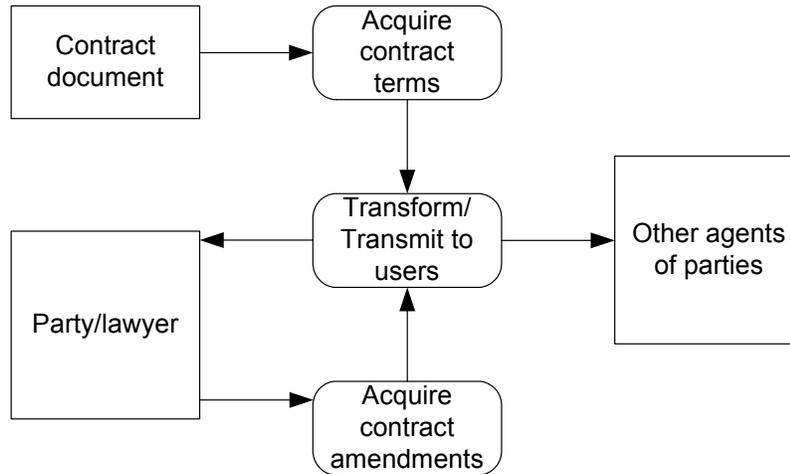
Information cannot be extracted from the contract document automatically unless that information is in a machine readable form. For all practical purposes, few, if any contract documents produced today under a negotiated contract terms process provide machine readable information. Most contract documents are either printed on paper or are held in an electronic format that does not provide any way to reliably extract meaningful information about the contract.

**Figure 5 Overview of transaction activity management**

### 3.3.7.3 Contract document management

This process is described in Figure 6.

Management of the contract document is important in transactions that run for a lengthy period with multiple persons who must perform obligations under the contract. Construction contracts are a good example. During the course of the project, there may be amendments to the contract terms that must be communicated to relevant agents of the parties. These persons may be operating under diverse conditions and may require access to the contents of contract documents in formats convenient to their circumstances. These include hard copy, searchable and browseable electronic copies, portable device access and even text to speech access.

**Figure 6 Managing the contract document**

### 3.3.8 Dispute resolution process

If a contract dispute arises, a formal dispute resolution process may be invoked by the parties.

Dr Leff proposed a use case that involves automated dispute resolution for electronic business contracts where the contract rights and obligations are expressed in machine readable form and there is no factual dispute between the parties. At this time there are no practical examples of this activity being undertaken or proposed. It is necessary to determine whether explicit support for that process is within the scope of the TC's current work.

### 3.3.9 Archive completed contract process

This step is shown for completeness. It raises no issues for the development of the TC's specification.

### 3.3.10 Data flows

At this stage, little is to be gained from a detailed analysis of the data flows. Sufficient information is provided by Figure 1 and the process analysis.

### 3.3.11 Interfaces

At this stage, little is to be gained from an analysis of specific interfaces. Sufficient information is provided by the process analysis.

## 3.4 Ticket contracts

Ticket contracts are those where a printed set of contract terms is offered to a buyer at or near to the time of purchase of a good or service. Examples include travel and parking station tickets.

This category also covers contracts such as “shrink wrap” software licenses that are included in physical software packaging.

The contract documents are normally prepared for the supplier by a process that is effectively the same as that described in Figure 2 (Contract drafting processes).

If the contracts are offered online in electronic form, ticket contracts are the same as click through contracts.

Printed ticket contracts raise no distinct issues relevant to the work of the TC and are not considered further in these requirements.

### 3.5 Standard form business and consumer contracts

This case describes contracts offered with a service in circumstances where the offeror will not accept negotiation of contract terms. Examples include housing finance, car financing and most insurance contracts. The offered services may be available from many suppliers on different terms. Consequently, the negotiation is effected by shopping around different service providers.

Unlike the ticket contracts, contract documents produced under this case bear more resemblance to negotiated contracts in the way they are produced. Some may be generated by document assembly systems to align the contract document with selected service options. Data values may be directly incorporated into or attached to the contract terms.

These contracts may be assented to in paper form or in an online transaction. If the contracts are offered online in electronic form, these contracts are the same as click through contracts discussed in the next section.

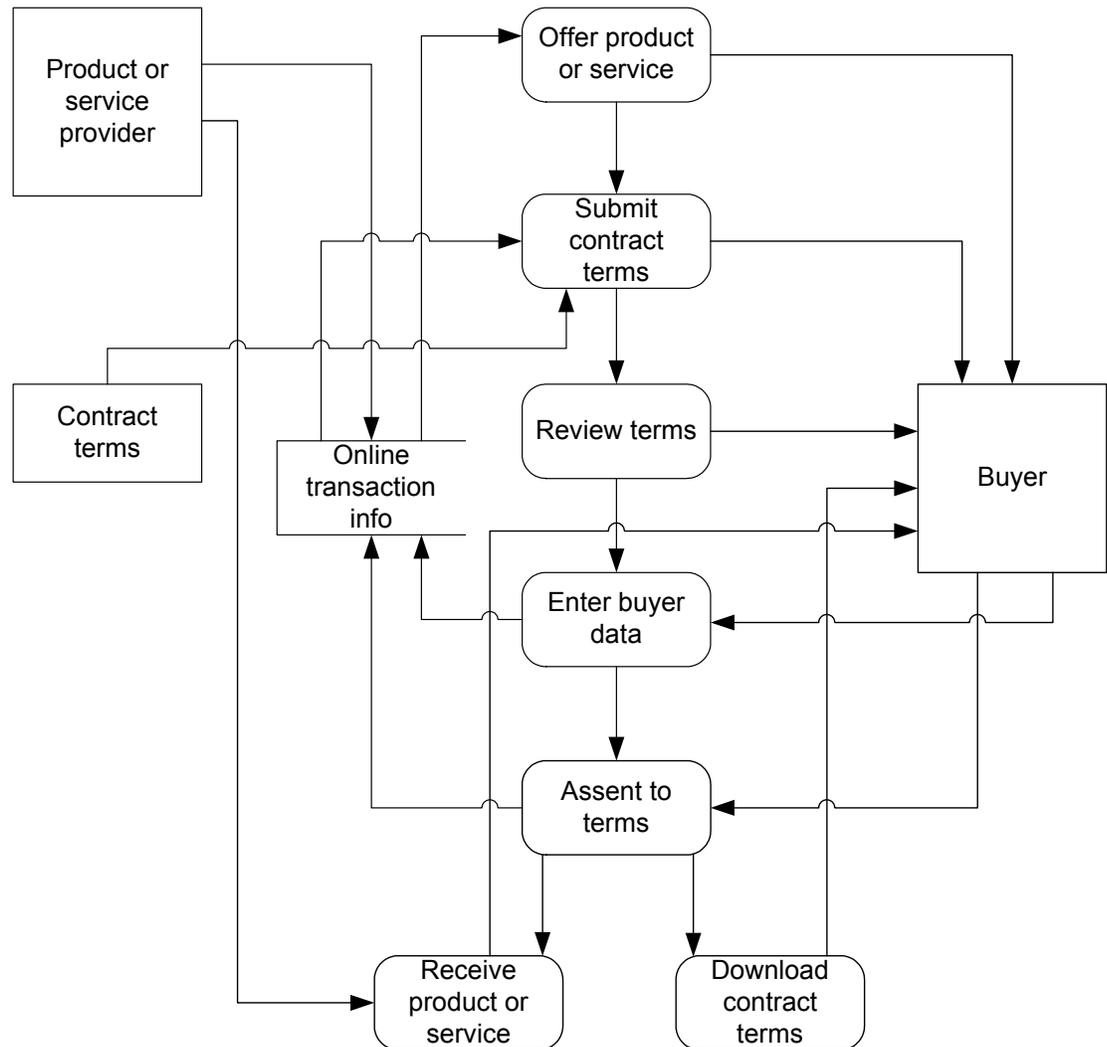
The standard form contract documents are normally prepared for the supplier by a process that is effectively the same as that described in Figure 2 (Contract drafting processes). These contracts raise no distinct issues for the TC's work.

### 3.6 Click through contracts

#### 3.6.1 Description

Click through contracts are really just electronic versions of ticket contracts or standard form business or consumer contracts. They involve a standard form contract document that is offered to a purchaser of goods or services during an on-line transaction or during software installation. The processes involved in an on-line click through contract are described in Figure 7.

Click through transactions are widespread on the World Wide Web.

**Figure 7 Click through contract overview**

### 3.6.2 Submit contract terms process

The service provider presents its standard contract terms for the selected service. The buyer is not able to negotiate these terms and must either accept the terms in full to receive the product or service or reject them and do without.

Some sites provide a way to access the standard contract terms at any time so they can be reviewed in advance, if desired. This is not invariable.

The contract terms presented are normally prepared for the service provider under a process that is basically the same as that described in Figure 2 (Contract drafting processes). The contract document is prepared once and used many times.

### 3.6.3 Assent to terms process

Assent is usually signified by the buyer clicking an “I agree” button or something similar on screen with a copy of the contract terms displayed. The

online transaction system records this event to initiate the next steps in the transaction.

#### 3.6.4 Download contract terms

The buyer may or may not have a convenient method to download a copy of the contract terms as a record of the transaction. Commonly, sites allow users to save a copy of the terms in HTML or PDF format and to print them. The document so created usually does not represent the complete terms of the contract between the parties. The transaction data values such as product, quantity, price and date are likely to be held in a database system and reproduced in a separate, automatically generated invoice or receipt.

#### 3.6.5 Other sub processes

At this stage it appears sufficient information is provided by this high level representation of click through transactions.

### 3.7 Electronic commerce contracts

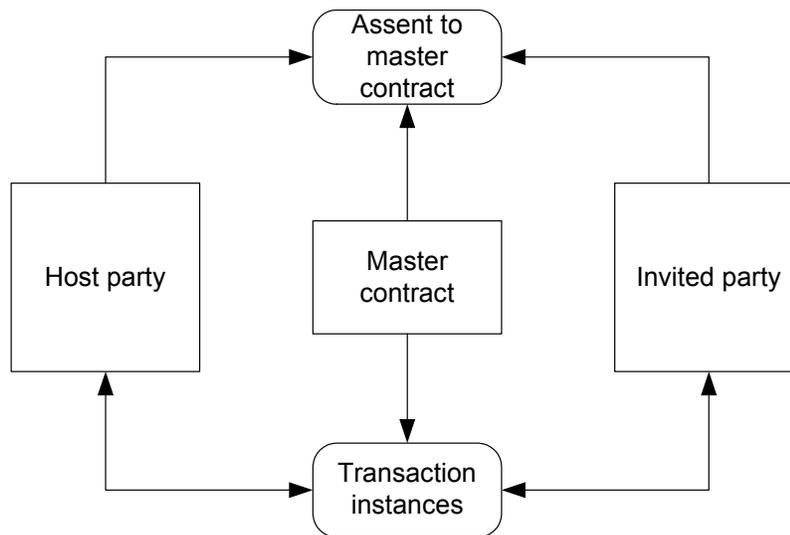
#### 3.7.1 Description

Commonly, electronic commerce contracts are set up between a large enterprise that wishes to do business electronically with its customers or suppliers for particular goods and services. Under this model, there is a host party that initiates the formation of the system and invited parties who must join the system and assent to the host's master contract if they wish to do business with the host.

Alternate models involve peer to peer exchanges under which all parties are essentially equal in bargaining power.

Electronic commerce transactions of this kind are not yet widespread. They are mainly confined to use by very large corporations such as aircraft or motor vehicle manufacturers and their suppliers.

An overview of the hosted model is described in Figure 8.

**Figure 8 Electronic commerce contracting system overview**

### 3.7.2 Assent to master contract process

The master contract is typically prepared by the host party and presented to the invited parties. In the early stages of the establishment of the system, it is likely that negotiation will occur over the terms of the master contract. This will take place outside the system.

The master contract is usually expressed in natural language and would be prepared in the same way as a draft contract described in Figure 2 (Contract drafting processes).

The parties could assent to the master contract by signing paper copies in much the same way as the negotiated contract terms scenario or by signifying their assent online in a similar way to the click-through contracts scenario. This may involve use of digital signatures.

### 3.7.3 Transaction instances process

Transaction instances occur when the parties wish to buy or sell a product or service provided by the master agreement. The master agreement specifies the procedures that must be followed through enquiry, offer and acceptance to create a binding transaction. The electronic trading system allows the parties to transmit machine readable messages to effect these transactions. Protocols for the machine readable messaging are provided by standards such as ebXML.

## 3.8 Computer negotiated contracts

Several researchers have proposed systems that will enable computer systems to automatically negotiate contracts to buy or sell commodity goods. These systems involve the specification of transaction parameters that can be understood by both sides to the negotiation.

The set of parameters that results from the negotiation must be mapped to a set of contract terms to provide a complete contract. This could occur simply by referring to a master agreement in the same way as for the electronic commerce case. Alternatively, the parameters may map to a set of contract terms derived from an agreed set of available terms in a way that is similar to a document assembly process for contract document drafting.

Processes involved in automated negotiation are outside the scope of the TC's work. Processes involved in relating contract parameters to contract terms is likely to be within scope.

## **4 The use of XML markup for contract documents**

### **4.1 The purpose of this section**

It is envisaged that various aspects of the TC's specification will be based on the use of XML for contract document preparation and contract management. This section provides a brief review of the key factors affecting the use of XML markup for contract documents.

### **4.2 The use of XML markup in contract authoring today**

As far as the TC is aware, almost all contract documents today are prepared in formats used by conventional word processing software such as Microsoft Word, Corel WordPerfect and other similar applications.

The TC has no evidence that contract drafters such as lawyers in law firms and enterprise legal departments are using XML authoring applications to create contract documents.

There is evidence that XML is used to markup precedent contract documents for use as back end XML in some document assembly systems. These systems must rely on the product vendor's proprietary XML schema since there are no applicable standard schema at this time. Authors interact with the precedents through an interface that shields them from the underlying document markup. Once the assembly process is complete, the document is converted to a word processing format so that any custom editing can be completed using common word processing applications familiar to the authors.

It is possible for users of Microsoft Office 2003 to create XML documents using the Microsoft WordML schema, provided that the enterprise concerned develops the necessary style mappings and import and export scripts. The WordML schema is a proprietary schema, owned by Microsoft.

It is also possible for users of OpenOffice ([www.openoffice.org](http://www.openoffice.org)) and StarOffice to create XML documents using the OpenOffice DTD. As with WordML, the OpenOffice DTD defines a format based XML markup that does not require the author to have any direct knowledge of the XML markup that is created during document authoring.

Neither WordML or OpenOffice XML are generic structural markup language schema. Because of the styling mechanisms they employ, both are essentially tied to particular software applications used to create them (Microsoft Word

and OpenOffice/Star Office). It is not the TC's role to promote a particular application suite but it may be necessary to determine how the TC's specification relates to those applications and XML vocabularies.

#### 4.3 Available tools for authoring contract documents using XML

There are commercial and open source XML authoring applications that can be configured to create contract documents using XML markup. Use of an XML authoring application requires a suitable schema or DTD and an application that will allow the contract document to be transformed into the desired layout formats such as RTF, PDF or HTML etc.

The TC is not aware of any public standard XML schema that are specifically designed for marking up contract documents. All available schema would require substantial modification to be useful.

XML authoring applications must work with many different schemas or DTDs. Many of the authoring conveniences found in word processing applications are not available out of the box with XML authoring applications. Usually, it is necessary to undertake a great deal of configuration development to create a tool that is convenient for authors to use. This configuration is usually specific to each DTD or schema.

Based on currently available tools, this development requires specialised expertise and often involves considerable effort and expense.

#### 4.4 Current use of contract metadata

Contract metadata is used extensively to categorise content for retrieval purposes, to manage versions and to support publishing processes.

Word processing applications make it difficult to associate metadata with individual content chunks.

XML markup makes it easy to add metadata to any chosen component. Metadata is commonly used with generic structural markup in XML documents.

#### 4.5 Current use of embedded data value markup for contracts

Embedded data value markup is used with word processing documents by common document assembly and variables substitution applications. In the absence of standards, these applications use proprietary formats for the definition of embedded data values.

The use of embedded data value markup is confined to document creation from precedent documents. Once the document is created, the embedded values are likely to be indistinguishable from other natural language content. The TC is not aware of applications that extract embedded data values from word processor prepared contract documents.

## 4.6 Machine generated XML markup

It is possible to generate a contract document using XML markup, either from another XML document or from chunks of text in a database. This approach is not useful for negotiated contracts where a lawyer or other contract author must modify the draft terms using commonly available desktop software.

## 4.7 The need to transform XML markup into display formats

Transformation or style rules must be applied to XML documents to enable them to be published in the desired format (RTF, PDF, HTML, etc) and according to the desired presentation layouts and styles. A variety of methods and tools exist for doing this. At this time, there is no universal approach that provides all the functionality required. The TC expects that for the foreseeable future, various enterprises may wish to use different tools to transform XML markup to display formats.

The desire to create multiple output formats from a single source is one of the key reasons why enterprises might use XML markup for documents. However, based on currently available tools, the planning and development of transformation rules or styles for a new DTD or schema requires specialised expertise and involves considerable effort and expense.

## 4.8 Exchange of XML contract documents between the parties

### 4.8.1 Exchange during negotiation

Opportunities to exchange XML documents between contracting parties or their representatives are limited firstly by the fact that contract drafters do not use XML authoring tools to any appreciable extent at this time.

If a party does draft a contract document using an XML editor, that party has a choice of providing the other party with the source XML, paper, PDF, HTML, RTF or possibly other formats that it can generate with available software. It would only provide an XML document if this is acceptable to the recipient.

Before assent, it is likely that the contract author (using an XML editor) may want the other party to make changes on the XML draft so it does not have to re-enter those changes from another format back into the XML document. This is a significant practical problem for the original contract drafter.

A party who receives an XML contract document for editing purposes must be able to conveniently use it with software they have available. The parties must work with a common schema so that the document is valid in both applications. The recipient must be able to format the document for editing and review purposes (not necessarily using the other party's styles). The recipient would like to be able to use automatic numbering and cross references.

### 4.8.2 Exchange during assent

Exchange of XML data at assent may occur in automated electronic business systems. The XML exchanged is likely to be quite different to that created for conventional contracts with negotiated terms.

### 4.8.3 Exchange after assent

Outside of automated electronic business systems, there is no obvious way in which an XML document will become the authoritative contract document. Normally, assent is signified by the parties signing a paper version. Less commonly, an electronic document may be adopted but this is likely to be a PDF document or something similar, rather than an XML document.

This means that even if the contract document was created in XML, that document is no more relevant to the resulting contract than a Word document is to the paper contracts signed in everyday transactions today. The XML document may or may not accurately reflect the final terms of the contract, depending on events after the assent version was prepared.

## 4.9 Likely future developments

Many new standards and products are being developed that may affect the way XML may be used for document production.

On the evidence available to the TC, it is apparent that law firms and other enterprises with corporate legal departments are not demanding to create documents using XML authoring tools. On the contrary, they need to be convinced there are good reasons to do so. They need to be convinced that suitable tools exist to make it easier, rather than harder, for authors to create documents to justify undertaking the change management and development effort.

Microsoft has introduced XML capabilities into Word 2003. It is extending the tools available for the use of XML and it is likely to make further changes to its Office applications in future releases. This will greatly affect future directions because it directly affects the tools most contract authors use today.

At this time it is difficult to forecast any widespread use of XML for contract authoring, except as back end XML described earlier.

## 4.10 Conclusions on the use of XML markup for contract documents

- 4.10.1 There is little likelihood that contract authors will prepare documents using XML authoring applications in the near future. They will continue to create contract documents using word processing applications and exchange word processing, PDF or printed documents.
- 4.10.2 There is no established infrastructure in place for parties to exchange and process XML contract documents. This situation is unlikely to change in the near future.
- 4.10.3 Even where back end XML is used to facilitate document assembly and other automated processing, there is no immediate prospect that contract documents in XML format will be distributed outside the enterprise.
- 4.10.4 Metadata and embedded data value markup can be utilised at the contract drafting stage by enterprises that use back end XML systems but this markup can only be available to other parties or interested persons if there is an exchange of XML documents.

## 5 The use of deontic contract languages

### 5.1 Review of current use

The purpose of deontic contract languages is to provide machine readable ways to express policy-based contract constraints. A system that can reason about obligations, rights, permissions, prohibitions, delegation, authority and similar concepts that underpin contracts could assist with contract management activities such as performance monitoring.

Three examples of deontic contract languages have been brought to the attention of the TC. One example is the Business Contract Language (**BCL**) developed at Distributed Systems Technology Centre, University of Queensland. The second example is the Contract Expression Language (**CEL**) developed by the Content Reference Forum ([www.crforum.org](http://www.crforum.org)). The third example is the Enterprise Contract Language (**ECL**) developed at the University of Kent. The ECL is conceptually similar to the BCL and is not further discussed at this time.

The “BCL has been developed for the purpose of specifying contract conditions so that the contract execution can be monitored against these conditions” [Zoran Milosevic]. BCL syntax is said to closely resemble natural language expression of contracts, namely the expression of deontic constraints such as obligations, permissions and prohibitions. The BCL is expressed using XML. It is a declarative language to be used by domain experts to specify those constraints that are required to be interpreted by a computer system.

Currently, there are no commercial implementations of the BCL.

The CEL is also inspired by deontic logic. It is based on the MPEG Rights Expression Language and is also expressed in XML. The CRF is a consortium of hardware companies, software companies, content owners and service providers working to develop standards for the legitimate distribution of content (licensed intellectual property) in peered environments. The CEL is used to express the contractual terms between participants in unambiguous, machine readable form.

A description of the CEL and its conformance to the Business Collaboration Framework (BCF), developed by the Techniques and Methodologies Group (TMG) of the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) is contained in the white paper *Contract Expression Language (CEL) – An UN/CEFACT BCF Compliant Technology* <http://www.crforum.org/papers/CEL-BCF-Whitepaper.pdf>. This paper includes examples of the markup of simple contractual promises.

It seems likely that the CEL will gain commercial adoption by some or all of the CRF consortium members.

Zoran Milosevic reports that there are a number of dedicated contract management vendors, including DiCarta ([www.dicarta.com](http://www.dicarta.com)), Upside Contracts ([www.upsidecontracts.com](http://www.upsidecontracts.com)) and iMany ([www.imany.com](http://www.imany.com)). These vendors aim to support the full contract life cycle management, ranging from collaborative contract drafting and negotiation, storage, milestone driven notification and analytic features. These systems are said to generally follow the database approach typical of most ERP systems. The contract semantics is implicitly encoded in proprietary software systems. These systems are hard to maintain, difficult to extend to new contract models and they do not permit inter-organisational collaboration. Deontic languages such as BCL and CEL aim to overcome these problems.

Based on information available, it appears that deontic contract languages are still subject to ongoing research. However, there appears to be several communities anxious to achieve commercial implementations. It would appear that these languages are best suited to represent contracts that meet some or all these criteria:

- (a) contract execution is to occur over a period of time that justifies the effort of setting up a monitoring system;
- (b) the contract is one of many similar contracts that are highly standardised so that, even if contract execution is short, there is no overhead in formalising contract terms;
- (c) both parties have an interest in automated monitoring;
- (d) performance of obligations can be defined by events that can be monitored by a computer system.

## 5.2 Likely developments with deontic contract languages

There is no reason to doubt that successful implementations of deontic contract languages will be achieved in the future. In the first instance, usage is likely to be confined to relatively simple, highly standardised transactions such as content licensing proposed by the CRF.

It seems unlikely that deontic contract languages will be applied to contracts that require negotiation of contract terms with original natural language drafting. The application of deontic contract language markup to a contract is likely to be a difficult exercise for contract authors because specialised skills will be needed to create it. For a considerable time deontic contract languages are likely to be applicable only to narrow classes of contracts.

Examples of deontic contract language markup indicate that a natural language contract document and a machine readable contract document ought to be distinct documents. The deontic contract language does not need to encapsulate by markup the natural language terms in the contract. To do so would make the task of markup more difficult. If the natural language contract document is marked up using a generic structural markup language, the overlapping and interleaving of the two forms of markup would appear to be impracticable.

### 5.3 Conclusions on the use of deontic contract languages

5.3.1 There is no obvious basis on which the TC can choose to incorporate a particular deontic contract language into its specifications. At this time it is not clear whether it should attempt to do so.

5.3.2 There is no evident advantage in combining deontic contract markup language with the natural language contract document. Rather, it appears likely that deontic contract language markup will be more easily implemented if it is maintained separately from the human readable contract documents.

5.3.3 In the absence of immediate vendor support, it is not clear what role the TC can play to facilitate the adoption of deontic contract languages.

## 6 Problems, objectives and scope

### 6.1 Purpose of this section

This section firstly identifies the problems encountered by various users (interfaces) involved in processes identified earlier.

High level objectives are then defined to deal with those problems. These statements provide a reference point against which all requirements can be validated.

Finally, each contract type or process is assessed to determine its relevance to the TC's mission.

The problem and objective statements are based on the discussions at the face to face meeting in San Francisco in September 2004, with some minor adjustments.

### 6.2 Contract drafting and negotiation

#### 6.2.1 Reported problems

Its difficult to trace contract changes at the level of contract terms.

It is hard to re-use content from transaction documents, reducing knowledge management benefits and increasing costs.

The drafting cycle during negotiation is slow, adding to cost.

Lawyers are expensive and clients would like to be able to do more contract drafting themselves.

Precedent maintenance is expensive, thus limiting access to precedents, increasing risk and cost.

How to translate contract terms into machine readable language.

### 6.2.2 Problem statement

Today, contract documents are created using word processing applications. These documents can't easily be processed at convenient levels of granularity. It is difficult to process these documents outside the creating application. This inhibits automated document creation, information reuse, information extraction and change traceability.

### 6.2.3 Objectives

The TC specification will enable:

- (a) contract terms to be managed and manipulated at convenient levels of granularity to facilitate:
  - (i) improved access to precedent contract terms and contract documents
  - (ii) better searching and discovery of relevant contract terms
  - (iii) improved document assembly systems
  - (iv) easier harvesting of re-usable contract terms from transaction documents
  - (v) collaborative authoring and negotiation
  - (vi) Revision history management
- (b) consistent, automated, enterprise wide formatting of contract documents
- (c) contract documents to be automatically published in multiple output formats.

### 6.2.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 6.3 Click through contacts

### 6.3.1 Reported problems

End users don't know:

- (a) what contracts they have entered into;
- (b) their obligations under those contracts.

The service providers have no obvious interest in helping the buyer solve these problems.

All the systems for accessing the contract are provided by the service provider.

### 6.3.2 Problem statement

Parties to click through contracts cannot easily identify or manage the contracts they have entered into or the terms of those contracts.

### 6.3.3 Objectives

The TC specification will enable individuals and enterprises to:

- (a) identify click through contracts prior to assent
- (b) Store contracts assented to
- (c) identify the contracts and terms assented to, when dealing with supporting online service providers.

### 6.3.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 6.4 Contract management – transaction activities

### 6.4.1 Reported problems

There are frequent disputes over change authorisation in construction contracts and similar transactions where frequent variations occur.

It is difficult to ensure all parties have reliable information about upcoming obligations under the contract.

It is difficult to extract terms and embedded data values from the contract into content management systems.

It is difficult to access the content of external documents that are incorporated into the contract.

There is no reliable way to determine the state of contract events, obligations and processes.

It is difficult to monitor and analyse performance of parties over extended time periods.

### 6.4.2 Problem statement

At the moment, parties to contracts and other users of contract information have no way to exchange machine readable contract information for importing into contract management systems.

### 6.4.3 Objectives

The TC specification will enable:

- (a) parties to contracts to exchange machine readable contract information, particularly embedded data values, for importing into contract management systems
- (b) machine readable contract information to be associated with relevant human readable contract terms.

### 6.4.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 6.5 Contract management – contract document

### 6.5.1 Reported problems

Interested persons don't have access to the content of the contract in a convenient form (paper, RTF, PDF, web, text to speech etc).

It is difficult to ensure that all interested persons have access to fully up to date versions of contract documents.

### 6.5.2 Problem statement

At the moment, it is not practicable to provide interested persons with the content of contract documents in different formats that suit their access needs.

### 6.5.3 Objectives

The TC specification will enable parties to contracts and authorised agents of the parties to access the content of contract documents in formats convenient to them where the contract document is prepared using XML markup and maintained in an up to date form as amendments are made.

### 6.5.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 6.6 Electronic commerce contracts

### 6.6.1 Reported problems

Electronic commerce standards such as ebXML provide for only part of the contract transaction. The master agreements or transaction protocol agreements are only available in human readable form.

There is no way for new parties to automatically assent to the master agreement.

## 6.6.2 Problem statement

In electronic commerce transactions:

- (a) there is currently no way to map or validate electronic transactions against their master agreement
- (b) there may be no way to automatically determine if there is a master agreement
- (c) human negotiation of bi-lateral master agreements is too time consuming.

## 6.6.3 Objectives

The TC specification will enable:

- (a) electronic commerce systems to establish machine readable master contracts that will permit automatic validation of subordinate electronic transactions
- (b) automatic contractual opt-in to participate in the system
- (c) (consider whether the eContracts standard can provide a lingua franca for processing data values created from various e commerce standards such as ebXML etc).

## 6.6.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 6.7 Computer negotiated contracts

### 6.7.1 Reported problems

Contract negotiation is slow and expensive.

Human contract negotiation limits the value to the parties, particularly where many parameters are involved.

Currently, there is no way to ensure that both parties to the negotiation can generate identical contract documents from the negotiated parameters.

### 6.7.2 Problem statement

There is no standard way to map negotiated contract parameters to contract terms.

### 6.7.3 Objectives

### 6.7.4 Scope assessment

[Assess whether this is within the scope of the TC's specification]

## 7 Overview of the proposed specification

The TC aims to develop a specification that can be used by the widest range of users at all stages of the contract life cycle. The specific needs of users and the systems they use are diverse. There is only very limited use of XML for document markup at present and processing systems are immature. The proposed specification will set out to achieve core, common objectives with minimal prescriptiveness. Feedback from use of the specification will guide its future development.

The specification will define an XML conforming schema for the generic, structural markup of contract documents. This will include a framework to add contract metadata and embedded data values markup to suit particular contract transactions. It will be up to interested industry sectors to define particular semantic XML vocabularies for metadata and embedded data values markup relevant to those industry sectors.

The generic, structural markup will support all document creation and publishing processes described earlier.

For the foreseeable future, the TC expects that structural markup will be used mainly inside contract authoring enterprises and that there will be little, if any exchange of XML contract documents between parties or other interested persons. Where exchange of XML document does occur, it is as likely that it will be based on industry specific schema, the Microsoft WordML schema or the OpenOffice schema as on the schema developed by the TC.

The structural markup schema will facilitate the use of XML markup in contract documents that can be exchanged between the parties. It can be adopted to support contract management requirements as the relevant exchange protocols and processing systems are implemented by software developers and the parties.

The initial version of the specification will define normative components only as far as necessary to promote support from software vendors.

The specification will define protocols for the exchange of contracts metadata and embedded data values between the parties and other interested persons for contract management purposes. These protocols make no assumptions about the use of XML markup for contract documents or the schema used for XML markup of those documents. It will be up to the parties to extract this information from database systems, XML documents or to create the exchange data manually, according to the systems available.

Exchange protocols for metadata and embedded data values will be of limited utility unless implemented by contract management system vendors. The TC does not have any contract management vendor representation. This part of the specification will be non normative. Dr Leff proposed that such features might be marked as "for experimental use and preliminary adoption".

The specification will define protocols for the use of deontic contract language markup separately from the generic structural markup of the contract documents and for the exchange of deontic contracts language markup between interested parties. Again, the TC does not have any contract management vendor representation. This part of the specification will be non normative.

## **8 Functional requirements**

### **8.1 Approach to requirements development**

The following table summarises how the processes discussed in section 3 relate to each relevant contract type. A check mark is shown under a contract type if the process can occur with that contract type, although it may not be common. The points shown by the check marks are the key intersections for which requirements are to be considered.

**Functional processes and contract types**

	Negotiated	Standard form	Click through	E commerce master	Computer negotiated
Processes					
(1) precedent creation, storage & retrieval	✓	✓	✓		✓
(2) document assembly for new drafts	✓	✓	✓		✓
(3) variables substitution in new drafts	✓	✓	✓		✓
(4) custom authoring of contract terms	✓			✓	
(5) collaborative editing of draft contract terms	✓				
(6) publish draft contract documents in multiple output formats	✓	✓	✓		
(7) manage document versions during drafting	✓				
(8) exchange draft contract documents with other parties prior to assent	✓	✓	✓		
(9) produce the assent document	✓	✓	✓		
(10) assent to contract terms	✓	✓	✓	✓	
(11) retain or preserve assent documents	✓	✓	✓	✓	✓
(12) extract contract states, party obligations and rights into exchange format	✓	✓		✓	
(13) communicate contract obligations, rights and states to interested persons	✓	✓			
(14) prepare variations to contracts	✓				
(15) maintain variations and contract versions	✓				

	Negotiated	Standard form	Click through	E commerce master	Computer negotiated
(16) publish contract terms and related information to interested persons	✓				
(17) validate electronic commerce transactions against the transaction protocol agreement				✓	
(18) map negotiated contract parameters to contract terms					✓

### Drafting note

The TC needs to decide if the listed processes are adequate or within scope.

Requirements will be stated once, in the first process in which they arise. Later processes that would support an existing requirement will refer to the applicable requirements in a note.

## 8.2 Precedent contract documents

### 8.2.1 Overview

There is nothing distinctive about precedent contract documents compared to other precedent documents prepared by lawyers.

The systems used for contract documents should be applicable to other kinds of precedent documents.

### 8.2.2 Precedent creation, storage & retrieval

#### 8.2.2.1 Background to requirements

If precedent documents are to be maintained in XML format, users will need to be able to convert word processing documents into XML markup.

The key problem identified in relation to precedent documents in section 6.2 is the lack of ability to work with word processing text content in various levels of granularity to suit automated processing needs.

Precedents may be created to work with specialised document assembly software. The TC is not attempting to analyse these in depth or to define a standard for the way in which that software operates.

#### 8.2.2.2 Issues for contracts with negotiated terms

Precedent documents are derived from transaction documents and enhanced by subject area experts to provide reliability and flexibility for users. Precedent documents must be regularly updated. New precedent terms are added as transactions evolve and new experiences encountered.

#### 8.2.2.3 Issues for standard form contracts

A standard form contract may be managed as a complete document or as a collection of discrete terms that can be assembled into a contract document to reflect transaction options. This does not appear to raise any issues different from those applicable to precedents for negotiated contracts.

#### 8.2.2.4 Issues for click through contracts

As for standard form contracts.

#### 8.2.2.5 Issues for computer negotiated contracts

It is understood that the precedent terms would need to be accessible in a way that a system could access them by reference or copy them into a contract document instance. Current understanding is that it will be sufficient if contract terms can be stored and retrieved as distinct objects.

#### 8.2.2.6 The role of XML markup

XML markup of precedent documents would provide these benefits:

- (a) It reduces the risk of obsolescence of valuable data as software is updated or replaced.
- (b) It enables content to be managed as convenient components that can be stored, retrieved and shared separately by software systems.
- (c) It enables documents to be automatically transformed into any desired output format according to standard enterprise styles.

There is no evident difference in the needs of the markup for each contract type considered.

#### 8.2.2.7 The role of a standard

A standard would provide these benefits:

- (a) The cost of implementation of XML based authoring and publishing applications could be reduced if applications are developed around a standard schema.
- (b) The costs of switching between document assembly and other processing software applications would be reduced if those applications are developed around a standard XML schema. However, enterprises may still incur some costs in adapting precedents to suit the more specialised interfaces of document assembly applications if they are not standardised.
- (c) Document assembly systems should be accessible to more enterprises who create contract documents.
- (d) It may be easier to develop and maintain processes to convert word processing data to XML markup if the target is standardised.

#### 8.2.2.8 Specific requirements

**R-1** The specification must include an XML schema for generic, structural markup of contract documents.

**R-2** The generic, structural markup schema must conform to W3C XML 1.0.  
*[note: We are not specifying here whether the schema is expressed in RELAX NG, XML Schema or DTD syntax.]*

**R-3** The generic, structural markup schema must define common content objects such as paragraphs and clauses that may be processed as distinct objects or content chunks in document assembly or other processing systems.

**R-4** It must be an aim in design of the generic, structural markup schema that the common content objects defined by the schema could be adopted by another standards body responsible for developing a schema applicable to other legal documents that are commonly prepared by the same people using the same systems as contract documents.

**R-5** The generic structural markup schema must provide a model for users to add semantic metadata and embedded data values to contract documents and to distinct content objects defined by the schema. The schema must make provision for common metadata fields required by document management, document assembly and publishing applications such as:

- (a) document identifiers, the author, version and dates;
- (b) the legal subject matter or categorisation of distinct content objects.

The schema must not restrict the metadata that users may add to contract documents or content objects.

The specification will not define a legal classification vocabulary for metadata values. These may be developed as required by industry or regional users.

**R-6** The specification must not require the use of any particular XML processing technology. As far as practicable, it must be designed to allow users to adopt any processing technology of their choosing.

## 8.3 Contract document creation

### 8.3.1 Document assembly for new drafts

#### 8.3.1.1 Background to requirements

There are various approaches to document assembly. Essentially, the process involves a software application presenting a drafter with choices about the terms to be included in a draft contract or other document. As the user signifies his or her choices, the application builds a document to reflect those choices.

The proposed specification will not promote any particular approach to document assembly. Different applications will have particular requirements for markup in the precedent documents. While this may diminish the value of a standard to users, it is unavoidable at this time. If software vendors can work with the proposed generic, structural markup there will be significant advantages to user enterprises who use that schema.

It is theoretically possible that someone could use an XML document as an assent document. The writer is unable to conceive of a situation where this would occur at this time. It is assumed that, in all contract types, the XML precedent document is transformed into a display format after document assembly.

#### 8.3.1.2 Issues for contracts with negotiated terms

Typically, the document assembly process produces only a first draft contract document. The author must then modify the draft to tailor it to the particular transaction at hand. Since authors work with word processing applications, it is assumed that an XML precedent must be translated to a word processing format or that an XML editor must be used for further drafting. If this approach is to work, the resulting word processing document must be of the same standard as one created from scratch in a word processing application. For example, it must have named styles and automatic numbering.

### 8.3.1.3 Issues for standard form contracts

In this case, the document produced by a document assembly process will be the final document submitted for assent. It is assumed in this case that there is no negotiation over narrative terms of the contract.

In this case, it is necessary only to translate the XML precedent document into a format suitable for assent, for example print, PDF or HTML.

### 8.3.1.4 Issues for click through contracts

As for standard form contracts.

### 8.3.1.5 Issues for computer negotiated contracts

As for standard form contracts.

### 8.3.1.6 The role of XML markup

While document assembly processes are undertaken without XML, there are many disadvantages, as previously stated. XML provides the benefits described in section 8.2.2.6.

### 8.3.1.7 The role of a standard

The benefits of a standard are the same as those listed in section 8.2.2.7.

### 8.3.1.8 Specific requirements

Requirements R-1 to R-6 apply to this process.

**R-7** The generic structural markup schema must provide sufficient definition of content objects in contract documents that user applications can:

- (a) define and apply automatic numbering schemes to those objects;
- (b) automatically transform or style contract documents into chosen output formats, including but not limited to print, RTF, PDF, HTML and text to speech ready formats.

**R-8** The generic structural markup schema must enable users to include all content necessary to create a contract document that could be used as an assent document. This does not require that attachments such as plans and other documents created by other persons must be marked up using the schema.

## 8.3.2 Variables substitution in new drafts

### 8.3.2.1 Background to requirements

Variables substitution involves the insertion of data values into placeholders in the narrative text of a precedent document or contract.

Variables substitution may occur with or as a separate process to document assembly. When highly standardised documents are used, variables substitution with a precedent contract may be the only process needed to prepare a contract document.

### 8.3.2.2 Issues for contracts with negotiated terms

Typically, the contract author will enter or import data values into a database format suited to the available processing software. Precedent documents may contain named placeholder markers which are replaced by the data values. The software provides a facility for the author or a system administrator to map named placeholders to data values in the database. When the draft document is otherwise complete, the data values are inserted into the document.

This process does not require the use of XML. It is commonly performed with word processing documents using a range of different software applications. The proposed specification is not concerned with word processing based operations.

If precedent documents are maintained in XML format, it is possible that variables substitution may be performed on the XML document or on a word processor version after translation. The proposed specification is not concerned about how or when variables substitution is carried out.

In word processing formats, some software vendors can read placeholder markers from other software vendors. This mitigates the lack of standards but does not reduce the need for costly data conversion from time to time.

There would be advantages to user enterprises if the placeholder markers in XML documents are standardised to provide greater interoperability of precedent documents with software from multiple vendors. It is not clear if this is practicable at this time. Consequently, it is proposed that the specification for markup of placeholders is non normative in the first version.

The placeholders for variables substitution in precedent documents are essentially placeholders for embedded data values. If marked up in the XML, the values can be read out by downstream processing applications.

### 8.3.2.3 Issues for standard form contracts

The issues are essentially the same as for contracts with negotiated terms.

### 8.3.2.4 Issues for click through contracts

As for standard form contracts.

### 8.3.2.5 Issues for computer negotiated contracts

As for standard form contracts.

### 8.3.2.6 The role of XML markup

XML markup may provide a way to standardise placeholder variables in precedent documents, if supported by software vendors.

XML markup of placeholders in precedents would make it easy to automatically markup embedded data values for later retrieval in downstream processing applications.

### 8.3.2.7 The role of a standard

A standard would be advantageous, for reasons canvassed earlier. It would be difficult to expect compliance with so few software vendors represented on the TC. It is proposed that this part of specification will be non normative.

### 8.3.2.8 Specific requirements

Requirements R-1 to R-6 apply to this process.

**R-9** The specification must define one or more elements as placeholders in XML precedent contracts that can be used for variables substitution. Variables substitution includes the insertion of markup to permit the extraction of embedded data values from contract documents.

**R-10** It is necessary to determine how the specification should allow users to define the content that is required to be inserted for each placeholder.

## 8.3.3 Custom authoring of contract terms

### 8.3.3.1 Background to requirements

### 8.3.3.2 Issues for contracts with negotiated terms

### 8.3.3.3 E commerce master

### 8.3.3.4 The role of XML markup

### 8.3.3.5 The role of a standard

### 8.3.3.6 Specific requirements

**R-11**

**R-12**

**R-13**

### 8.3.4 From draft terms to authoritative contract document

#### 8.3.4.1 Background to requirements

Except for computer negotiated contracts, the first draft of the contract terms are unlikely to satisfy all parties. Where a party has sufficient negotiating power, it may negotiate amendments to those terms.

Once negotiation is complete, a document will be prepared for assent by the parties. After assent, that document will be the authoritative contract document. It is the only one that everyone can be certain reflects the final agreement between the parties.

The negotiated drafting process is central to contracts with negotiated terms. It is not relevant to standard form contracts or click through contracts.

It is not relevant to computer negotiated contracts, since in that case the first draft is assumed to be the end result.

#### 8.3.4.2 Issues for contracts with negotiated terms

One of the parties will take responsibility for preparing a draft contract document. Securing this responsibility provides a strategic advantage, since the wording of this initial draft contract document will then only change via negotiation.

The party's representative will prepare an initial draft contract document. The representative will be a lawyer or a contracts manager. If a lawyer, they may be an employee in the legal department, or a third party (ie a lawyer at a law firm).

In law firms, a precedent document with or without document assembly may be used as a starting point for the document, which will then be edited in Word (or some other word processor).

Contract managers in large corporates may use a contract management system to prepare the first draft contract document. These systems now often feature round-tripping to Word, and if so, the terms may be able to be edited in either environment.

In most cases, the representative will present the resulting draft contract document to his or her client for comment before presenting it to the other party or parties.

Typically, this is done by emailing the file in Word format. However, increasingly, the initial draft contract document is made available in a shared workspace. For example, access to a law firm's document management system via their extranet. This may involve publishing the document in other formats, such as HTML.

The advantage of doing this is are several:

- (a) there is one source for the current version
- (b) the client can access the document at their convenience, from anywhere
- (c) it is easier to encrypt the document in transmission.

In most cases today, the contract document is accessed and edited as a whole, rather than at the level of the individual clause. This is because non-XML word

processing representations of a contract document do not lend themselves to that sort of manipulation. [see section 6.2.2]

Once the client approves the document, it will be presented to the other parties, again, by email or by giving them access to a shared workspace.

Where the contract document is in a shared workspace, the client may see a more version than the other parties can access.

The draft contract terms are edited as the negotiation proceeds. Sometimes the edits are intended to reflect verbal discussions, while at other times, new written terms may be offered for incorporation into the draft that aren't wholly anticipated by prior verbal discussion.

Typically, the representative who prepared the first draft contract document will maintain ownership of the process by which changes are accepted into the draft.

#### 8.3.4.3 The role of XML markup

A generic structural markup language will identify individual clauses, which will enable contracts in shared workspaces to be accessed and manipulated at the clause level of granularity.

This in turn will enable:

- (a) clause level versioning
- (b) clause level audit trails / history
- (c) at-a-glance reporting on the status of a negotiation
- (d) clause level access control, meaning other parties can see a modified clause as soon as it is approved, rather than having to wait for the entire document to be approved
- (e) simultaneous editing of multiple clauses by different authors.

#### 8.3.4.4 The role of a standard

Today, the shared workspace is typically owned and managed by one of the parties to the contract. The other parties are unable or unwilling to entrust their own records and working notes to it.

For this reason, they may wish to maintain their own record of the negotiation in a tool of their own choosing, in parallel with record the other party maintains and shares in part.

One role of a standard would be to make it easier for this party to maintain their own record.

In spite of the benefits of shared workspaces with clause-level features, people may wish to be able to edit the contract in Word. There are various reasons for this:

- (a) the document can be accessed in a disconnected environment
- (b) it may be easier to perform certain operations eg add new clauses, move clauses around

- (c) old habits die hard.

A standard ought to make it easier to take a contract document which is in Word format and to merge it on top of an earlier draft in the shared workspace, or indeed to simply create a new project in the shared workspace.

There are at least two ways to think about the opportunity for standardisation here:

- (a) the first is that a generic structural markup language would provide a standard import/export format for complete contract documents into or out of clause-level shared workspaces
- (b) the second is to think in terms of the clause chunks, and to think of a standard for getting and putting those chunks. This might use technologies such as WebDAV/DeltaV, JCP 147 & 170, XML:DB and Montag.

The TC should adopt an approach that assists parties to manage this in either way.

#### 8.3.4.5 Specific requirements

Requirements R-1 to R-8 apply to this process, although R-4, R-5, R-6 and R-7(b) are not material to these processes.

- R-14** It must be possible for processing systems to attach unique identifiers to distinct content objects defined by the XML schema described in requirements R-1 and R-3.

#### 8.3.5 Publish draft contract documents in multiple output formats

##### 8.3.5.1 Background to requirements

##### 8.3.5.2 Issues for contracts with negotiated terms

##### 8.3.5.3 Issues for standard form contracts

##### 8.3.5.4 Issues for click through contracts

##### 8.3.5.5 The role of XML markup

##### 8.3.5.6 The role of a standard

### 8.3.5.7 Specific requirements

#### **R-15**

#### **R-16**

#### **R-17**

### 8.3.6 Manage document versions during drafting

#### 8.3.6.1 Background to requirements

A contract document is likely to go through various iterations before it is ready, whether it is a contract with negotiated terms, a standard form contract, or a click through contract.

There are several differences between a contract with negotiated terms and the other forms:

- (a) in the case of a contract with negotiated terms, negotiation drives and constrains the iterations
- (b) in the case of a contract with negotiated terms, each party may wish to manage the document versions.

#### 8.3.6.2 Issues for contracts with negotiated terms

The person responsible for the drafting may manage the versions in any of the following ways:

- (a) ad-hoc retention of old versions ("File Sav As.."), possibly relying on their email outbox as a record of old versions
- (b) possibly using the version functionality in Microsoft Word (but unlikely)
- (c) using a document management system (eg Interwoven or Documentum or hummingbird)
- (d) using a contract management system.

In many cases, a lawyer will retain printed copies of old versions as well. These printed copies may well contain hand-written annotations.

Where they are using a document management system which can expose the contract in a workspace shared with their client, the client may rely on this to keep track of versions. Other parties are likely to maintain their own independent records, using one or other of the above strategies.

As described in 7.3.4, today versioning is likely to be at the level of the document rather than an individual clause (unless the drafting process is being managed within a contract management system), with the attendant disadvantages.

When harvesting clauses for later re-use via a knowledge management system, the latest version may not be the greatest, since it may bear the scars of negotiation and compromise. So one may need to harvest clauses from old versions. [see section 3.3.6].

### 8.3.6.3 The role of XML markup

A primary role of XML is the same as described in section 8.3.4.3, that is, to facilitate clause-level versioning. For this, generic structural markup is required.

However contract metadata is valuable for clause harvesting. Some metadata, such as author, is fundamental. If available, other metadata (such as type of clause (eg termination, indemnity), or precedent-value-assessment) would assist in automated clause harvesting. The problem here is that, even assuming the drafter uses an XML editor, this information is not likely to be added by lawyers during contract drafting. It adds more work but does not assist the drafting process.

Capturing contract metadata in the XML markup is not essential for clause-level versioning. That metadata - including the reason for the change - can be captured and stored using other means.

Clause-level contract metadata in XML markup, rather than just in a separate content management system enables the information to be communicated with the contract document or clause as it is transferred between systems.

### 8.3.6.4 The role of a standard

See section 8.3.4.4 regarding the role of a standard for general structural markup in facilitating clause-level versioning.

A standard for clause-level contract metadata would facilitate the development and market acceptance of knowledge harvesting clause libraries, since it would provide a standard format upon which those systems could operate, and increase their power. This is true even if XML contract documents are only used in small areas for the foreseeable future.

### 8.3.6.5 Specific requirements

Requirements R-1 to R-8 apply to this process, although R-4, R-5, R-6 and R-7(b) are not material to these processes.

**R-18** Clause-level metadata is required in order to facilitate the initial setup of a clause-level shared workspace. Not all of it would be exposed to other parties. Such metadata may include:

- (a) clause author
- (b) source (new, precedent, other party)
- (c) legal character (indemnity, warranty etc).

**R-19** Clause-level metadata is required in order to facilitate clause harvesting. Such metadata would include:

- (a) clause author
- (b) source (new, precedent, other party)
- (c) legal character (indemnity, warranty etc).

(d) precedent value assessment.

**R-20** The TC should make recommendations as to how clause level metadata may optionally be attached to clauses within Microsoft Word, for round-tripping with XML-based systems.

### 8.3.7 Exchange draft contract documents

#### 8.3.7.1 Background to requirements

#### 8.3.7.2 Issues for contracts with negotiated terms

#### 8.3.7.3 Issues for standard form contracts

#### 8.3.7.4 Issues for click through contracts

#### 8.3.7.5 The role of XML markup

#### 8.3.7.6 The role of a standard

#### 8.3.7.7 Specific requirements

### **R-21**

### **R-22**

### **R-23**

### 8.3.8 Produce assent document

#### 8.3.8.1 Background to requirements

#### 8.3.8.2 Issues for contracts with negotiated terms

#### 8.3.8.3 Issues for standard form contracts

8.3.8.4 Issues for click through contracts

8.3.8.5 The role of XML markup

8.3.8.6 The role of a standard

8.3.8.7 Specific requirements

**R-24**

**R-25**

**R-26**

8.4 Assent by parties to the contract

8.4.1 Assent to contract terms

8.4.1.1 Background to requirements

8.4.1.2 Issues for contracts with negotiated terms

8.4.1.3 Issues for standard form contracts

8.4.1.4 Issues for click through contracts

8.4.1.5 Issues for e commerce master contracts

8.4.1.6 The role of XML markup

8.4.1.7 The role of a standard

8.4.1.8 Specific requirements

**R-27**

**R-28**

**R-29**

**8.5 Contract management – contract activities**

**8.5.1 Extract contract states, obligations, rights etc into exchange format**

**8.5.1.1 Background to requirements**

**8.5.1.2 Issues for contracts with negotiated terms**

**8.5.1.3 Issues for standard form contracts**

**8.5.1.4 Issues for e commerce master contracts**

**8.5.1.5 The role of XML markup**

**8.5.1.6 The role of a standard**

**8.5.1.7 Specific requirements**

**R-30**

**R-31**

**R-32**

**8.5.2 Communicate contract states, obligations & rights to interested parties**

**8.5.2.1 Background to requirements**

**8.5.2.2 Issues for contracts with negotiated terms**

8.5.2.3 Issues for standard form contracts

8.5.2.4 The role of XML markup

8.5.2.5 The role of a standard

8.5.2.6 Specific requirements

**R-33**

**R-34**

**R-35**

## 8.6 Contract management – contract document

8.6.1 Retain or preserve assent document

8.6.1.1 Background to requirements

8.6.1.2 Issues for contracts with negotiated terms

8.6.1.3 Issues for standard form contracts

8.6.1.4 Issues for click through contracts

8.6.1.5 Issues for e commerce master contracts

8.6.1.6 Issues for computer negotiated contracts

8.6.1.7 The role of XML markup

8.6.1.8 The role of a standard

8.6.1.9 Specific requirements

**R-36**

**R-37**

**R-38**

8.6.2 Prepare variations to contract

8.6.2.1 Background to requirements

8.6.2.2 Issues for contracts with negotiated terms

8.6.2.3 The role of XML markup

8.6.2.4 The role of a standard

8.6.2.5 Specific requirements

**R-39**

**R-40**

**R-41**

8.6.3 Maintain variations and contract versions

8.6.3.1 Background to requirements

8.6.3.2 Issues for contracts with negotiated terms

8.6.3.3 The role of XML markup

8.6.3.4 The role of a standard

8.6.3.5 Specific requirements

**R-42**

**R-43**

**R-44**

8.6.4 Publish contract terms to interested persons

8.6.4.1 Background to requirements

8.6.4.2 Issues for contracts with negotiated terms

8.6.4.3 The role of XML markup

8.6.4.4 The role of a standard

8.6.4.5 Specific requirements

**R-45**

**R-46**

**R-47**

8.7 Other processes

8.7.1 Map negotiated parameters to contract terms

8.7.1.1 Background to requirements

8.7.1.2 Issues for computer negotiated contracts

8.7.1.3 The role of XML markup

8.7.1.4 The role of a standard

8.7.1.5 Specific requirements

**R-48**

**R-49**

**R-50**