

## OASIS LegalXML eContracts Technical Committee

### Discussion Paper - XForms for Contract Semantics

Jason Harrop, [jharrop@speedlegal.com](mailto:jharrop@speedlegal.com)

Version 1.0, Thursday, March 11, 2004

## **Introduction**

Many vertical industries have established XML schemas to represent the data which is of interest to them.

Examples include:

- insurance: ACORD ([www.acord.org](http://www.acord.org)) XML standards for life, property & casualty, and re-insurance
- financial products such as swaps/derivatives: ISDA's FpML
- real property
  - eg UK <http://www.pisces.co.uk/>
    - mortgage industry
      - eg MISMO's loan application DTD
      - eg Australia's <http://www.lixi.org.au/>  
<http://www.rets-wg.org/>  
Data Consortium
- intellectual property rights
  - eg ODRL (odrl.net, [www.iprsystems.com](http://www.iprsystems.com)),
  - eg ContentGuard's XrML ([www.xrml.org](http://www.xrml.org), <http://www.oasis-open.org/committees/rights/>)
- privacy
  - eg W3C's P3P
  - eg IBM's EPAL

In addition, OASIS's own Universal Business Language is an effort to markup business information which is common to many industries.

These schemas and the work of the OASIS LegalXML eContracts TC are complementary:

- These schemas capture a lot of the data making up a transaction (parameterise the problem domain), but they don't make use of any general standardised mechanism for relating that data back to the traditional signed legal document<sup>1</sup>.

---

<sup>1</sup>MISMO's "SMART" Document specification does contain a published mechanism for doing this,

- Meanwhile, one of the objectives of the eContracts TC is to provide a mechanism to markup semantic information. These schemas show what “semantics” are important in particular verticals, and thus what the eContracts TC needs to capture for its work to be relevant to those verticals.

The challenge for the eContracts TC is to find a way to use these existing schemas directly. This paper explores XForms as a possible approach.

### ***Why Structure Matters to Verticals***

As I've said, the schemas identified above capture a lot of the data making up a transaction. They tend to capture business terms, as opposed to more abstract legal concepts such as obligations, warranties, and grounds for termination.

They can be said to parameterise the problem domain, according to what is important to actors in the industry.

What these schemas don't do is provide any general standardised mechanism for relating that data back to the traditional signed legal document.

This means that the transaction data and the legal document could be inconsistent/contradictory, unless care is taken to ensure that whenever one is updated at any point in the life cycle, the other is as well.

Many industries operate using largely electronic processes on the basis of the data captured in the schema, and only turn to the terms of the contract when things go wrong.

This can be okay if the contracts are always identical (except for a schedule of data particular to each instance), but the reality is that even standard form contracts go through periodic revision.

When things go wrong, the last thing the legal department needs is a disparity between the data in your transaction system, and the terms of the contract, or to find that you've missed your chance to respond in some way.

In other industries and businesses, the contracts will be on paper. Someone has to go through them and rekey the data into the contract management system. The deal is under the radar until it is in the contract management system, and getting it there without transcription errors is another problem<sup>2</sup>.

Both of these problems arise because the transaction data and the legal document are handled according to different processes.

The promise of eContracts is to fuse the two together, so that there is never a disparity, and never a need for manual data entry.

### ***The eContracts Semantic Challenge***

Meanwhile, the eContracts TC has set itself the challenge of marking up semantic information.

---

but it has not been adopted outside MISMO's user community.

<sup>2</sup> See “Data Extraction and Conversion – Contract Management’s Dirty Little Secret”, <http://www.accruent.com/solutions/pubs/pdfs/whitepapers/ContractAbstraction.pdf>

For present purposes, **semantic information** can be thought of as information about the deal and/or of legal significance which can be discerned by perusing the contract itself.

This definition is intended to distinguish semantic information from metadata. **Metadata** pertaining to the contract is often not apparent from the contract itself. Metadata is information such as how many versions a clause has been through, who authored it, who approved it.

This paper is not about metadata.

There are two questions we need to ask ourselves regarding semantic information:

- the “what” question: what semantic information do we want to be able to capture?
- the “how” question: how are we going to do it?

A contract is full of semantic information. People with different backgrounds/expertise will identify different subsets of that information. For example, a lawyer is likely to pick out different things than a line manager in a corporation. And a tax specialist will drill down into different things to a litigation lawyer.

The granularity of the information differs, from word/phrase, to clause, or the interaction of several clauses.

Some of text is likely to be open to conflicting interpretations as to how it should be marked up semantically. There is also a fine line between fact and inference.

So we'll only have answered the “what” question satisfactorily, if we can also answer a “why” question convincingly.

It is only worth marking up a piece of semantic information if someone is going to use it. It will only get marked up by someone who has a direct interest in doing so.

The vertical industry schemas are handy evidence of what is important in those industries. Moreover, they are already in a convenient XML format!

It is quite possible that there is other semantic information which will prove valuable to people not directly served by the vertical schemas. For example, information the legal department may wish to capture. We may need to capture this as well.

However, by starting with the vertical industry schemas, our work is much more likely to be relevant to those verticals.

So a partial answer to the “how” question is to find a way to use these existing schemas directly. If we don't, our work will be that much harder to adopt, since it will be unfamiliar, and require conversion steps.

## ***Business Requirements***

Summary of business requirements identified:

- an XML representation of the contract which can simultaneously contain structural and semantic markup
- a representation which can use existing vertical industry schemas for the semantic markup
- attach semantic markup to phrases
- attach semantic markup to clauses
- attach semantic markup to groups of clauses

- ability to markup items of semantic interest not anticipated in a vertical industry schema

## ***Candidate Solutions***

The ability to make direct use of existing vertical industry schemas cuts out a lot of possible approaches to semantic markup.

For example, it knocks out the approach of having an extensive set of semantic markup which simply appears in the structural markup (eg in the <p>, or as an attribute on the <section> elements from XHTML 2).

However, there are three candidate solutions which meet that requirement and each have some existing market momentum.

These are:

- W3C's XForms
- Microsoft's InfoPath<sup>3</sup>
- Adobe Systems' Form Designer

Of these three solutions, XForms is clearly the most open. Not only is it a W3C Recommendation, but there are also many implementations already (see further below).

It is also a natural complement to our approach of using XHTML2 for structure, since XForms and XHTML2 are both part of the W3C's world view.

MISMO's "SMART" Document specification is also worthy of note. It has a section for data, and another for a "view", the view being "original, legal, and authoritative version" of the document. The view can either be a "tagged" document (XHTML), or an image (PDF, TIFF). The data section can include a map element, which contains arcs mapping elements in the data to corresponding parts of the view.

MISMO's "SMART" Document specification is directed at the need identified in this paper, but it doesn't have the market momentum of XForms.

## ***What are XForms?***

XForms is intended to replace HTML forms. XForms became a W3C Recommendation on 14 October 2003.

For our purposes, the key idea they bring to the table is a separation between the data the form is collecting, and the appearance of the form itself.

It is this idea, and the XForms implementation of it, which we can use.

## ***Example 1 – Simple Contract***

Consider the following very simple contract:

In consideration of the payment of \$1000, Widget Builders Inc agrees to deliver a red widget to Cautious Client Inc on or before 30 June 2004.

---

<sup>3</sup>For a comparison of XForms and InfoPath, see "XForms and Microsoft InfoPath" by Micah Dubinko at <http://www.xml.com/pub/a/2003/10/29/infopath.html>

Since the eContracts TC is basing its structural markup on XHTML 2, the structural markup for this is likely to look something like:

```
<html>
  <head>
    <title>Simple Contract</title>
  </head>
  <body>
    <section>
      <p>
In consideration of the payment of $1000, Widget Builders Inc agrees to deliver a red widget to Cautious Client Inc on or before 30 June 2004.
      </p>
    </section>
  </body>
</html>
```

Suppose the semantics we wish to capture relate to the words underlined:

In consideration of the payment of \$1000, Widget Builders Inc agrees to deliver a red widget to Cautious Client Inc on or before 30 June 2004.

Suppose further that there was simple vertical market schema that could be used to create an XML document with the following structure:

```
<contrived-semantics>
  <supplier/>
  <customer/>
  <lineitems>
    <item>
      <name/>
      <price/>
    </item>
  </lineitems>
  <deliverydate/>
</contrived-semantics>
```

Then an eContract XForm capturing both the structure and these semantics would be the following:

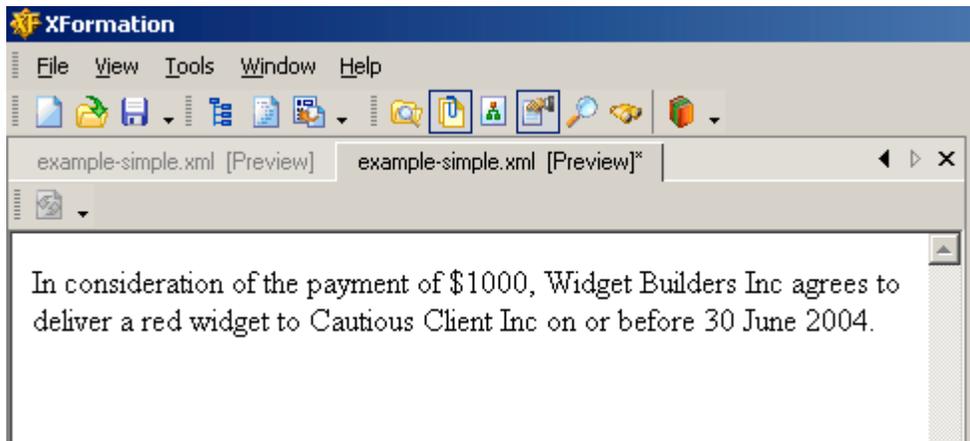
```
<html xmlns:xforms="http://www.w3.org/2002/xforms">
  <head>
    <title>Simple Contract</title>
    <xforms:model>
      <xforms:instance xmlns="">
        <contrived-semantics>
          <supplier>Widget Builders Inc</supplier>
          <customer>Cautious Client Inc</customer>
          <lineitems>
            <item>
              <name>red widget</name>
              <price>$1000</price>
            </item>
          </lineitems>
          <deliverydate>30 June 2004</deliverydate>
        </contrived-semantics>
      </xforms:instance>
    </xforms:model>
  </head>
  <body>
    <p>
```

In consideration of the payment of `<xforms:output ref="lineitems/item/price"/>`, `<xforms:output ref="supplier"/>` agrees to deliver

a

```
<xforms:output ref="lineitems/item/name"/> to <xforms:output ref="customer"/> on or before <xforms:output ref="deliverydate"/>.
  </p>
</body>
</html>
```

The result is as follows:



Obviously this is just a simple single paragraph example. There's no reason the contract couldn't be a formal document of say 30 pages in length.

That preview is in the XFormation IDE, which uses FormsPlayer. It would look the same in Internet Explorer.

Note the viewing the contract directly in the browser like that is just one way of looking at a contract document. Instead it may be transformed to XHTML (without XForms) or WordML or SVG, or some nonXML format such as PDF or RTF.

## ***XForms – how it works***

You can see the data captured by the vertical industry schema sits in an `<xforms:instance>` element inside the `<xforms:model>` in the XHTML `<head>`.

The eContracts structural markup appears inside the XHTML `<body>` element. - nothing new here.

Where there is semantic information which needs to be marked up, it is enclosed in an XForms `<output>` element.

(The XForms `<output>` element is a form control which renders a value from the instance data, but provides no means for entering or changing data, which is what we want. )

The output element is bound to the relevant part of the instance using `@ref`.

Note also that in our basic usage of XForms technology, the eContract doesn't actually have any visible form fields or a submit button.<sup>4</sup>

## ***Tool Support***

An advantage of using XForms is that there is a maturing body of tools available for eContracts software developers and end users alike.

For details, see <http://www.w3.org/MarkUp/Forms/> and "Ten Favorite XForms Engines" by

---

<sup>4</sup> If it did have a submit button, then hitting "submit" would submit/save the vertical industry data, rather than all the text of the contract contained in the eContract structure.

[Micah Dubinko](http://www.xml.com/pub/a/2003/09/10/xforms.html) at <http://www.xml.com/pub/a/2003/09/10/xforms.html>

The examples in this paper were created in JEdit, or using the XFormation editor (Windows only) and then viewed/tested using FormsPlayer (the Internet Explorer plugin), or X-Smiles.

## Example 2 – Simple Contract with UBL

Of course, the point of this approach is to leverage existing schemas.

So I shouldn't have made up my own semantics in example 1. Lets see what it looks like if we use UBL components.<sup>5</sup>

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:xforms="http://www.w3.org/2002/xforms" xmlns:ev="http://www.w3.org/2001/xml-
events" xmlns:cat="urn:oasis:names:tc:ubl:CommonAggregateTypes:1.0:0.70">
  <head>
    <title>Simple Contract</title>
    <xforms:model>
      <xforms:instance xmlns="">
        <Order xmlns="urn:oasis:names:tc:ubl:Order:1.0:0.70"
xmlns:cat="urn:oasis:names:tc:ubl:CommonAggregateTypes:1.0:0.70"
xmlns:cct="urn:oasis:names:tc:ubl:CoreComponentTypes:1.0:0.70"
xmlns:ccts="urn:oasis:names:tc:ubl:CoreComponentParameters:1.0:0.70" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:oasis:names:tc:ubl:Invoice:1.0:0.70 UBL_Library_0p70_Order.xsd">
          <cat:ID>20031234-1</cat:ID>
          <cat:IssueDate>2003-02-02</cat:IssueDate>
          <cat:LineExtensionTotalAmount currencyID="USD">1000</cat:LineExtensionTotalAmount>
          <cat:BuyerParty>
            <cat:ID/>
            <cat:PartyName>
              <cat:Name>Cautious Client Inc</cat:Name>
            </cat:PartyName>
            <cat:Address>
              <cat:ID/>
              <cat:Street/>
              <cat:CityName/>
              <cat:PostalZone/>
              <cat:CountrySub-Entity/>
            </cat:Address>
            <cat:BuyerContact>
              <cat:ID/>
              <cat:Name/>
            </cat:BuyerContact>
          </cat:BuyerParty>
          <cat:SellerParty>
            <cat:ID/>
            <cat:PartyName>
              <cat:Name>Widget Builders Inc</cat:Name>
            </cat:PartyName>
            <cat:Address>
              <cat:ID/>
              <cat:Street/>
              <cat:CityName/>
              <cat:CountrySub-Entity/>
            </cat:Address>
          </cat:SellerParty>
          <cat:OrderLine>
            <cat:BuyersID>1</cat:BuyersID>
            <cat:SellersID/>
            <cat:LineExtensionAmount currencyID="USD">1000</cat:LineExtensionAmount>
            <cat:Quantity unitCode="PKG">1</cat:Quantity>
            <cat:Item>
              <cat:ID/>
              <cat:Description>red widget</cat:Description>
              <cat:SellersItemIdentification>
                <cat:ID>32145-12</cat:ID>
              </cat:SellersItemIdentification>
            </cat:Item>
          </cat:OrderLine>
        </Order>
      </xforms:instance>
    </xforms:model>
  </head>
</html>
```

---

5I'm not the first to combine UBL with XForms. See "UBL in XForms, a worked example", at <http://xformsinstitute.com/ubl/>

```

        </cat:SellersItemIdentification>
        <cat:BasePrice>
          <cat:PriceAmount currencyID="USD">1000</cat:PriceAmount>
        </cat:BasePrice>
      </cat:Item>
    <cat:DeliveryRequirement>
      <cat:ID/>
      <cat:DeliverToAddress>
        <cat:ID/>
        <cat:Street/>
        <cat:CityName/>
        <cat:PostalZone/>
        <cat:CountrySub-Entity/>
      </cat:DeliverToAddress>
      <cat:DeliverySchedule>
        <cat:ID/>
        <cat:RequestedDeliveryDate>2004-06-30</cat:RequestedDeliveryDate>
      </cat:DeliveryRequirement>
    </cat:DeliveryRequirement>
  </cat:DeliverySchedule>
</cat:DeliveryRequirement>
</cat:OrderLine>
</Order>
</xforms:instance>
</xforms:model>
</head>
<body>
  <p>
    In consideration of the payment of <xforms:output ref="cat:LineExtensionTotalAmount"/>, <xforms:output
    ref="cat:BuyerParty/cat:PartyName/cat:Name"/> agrees to deliver a
    <xforms:output ref="cat:OrderLine/cat:Item/cat:Description"/> to <xforms:output ref="cat:SellerParty/cat:PartyName/cat:Name"/>
    on or before <xforms:output ref="cat:OrderLine/cat:DeliveryRequirement/cat:DeliverySchedule/cat:RequestedDeliveryDate"/>.
  </p>
</body>
</html>

```

## Discussion – instance superset

In this example, the instance contains several pieces of information which aren't referenced from elsewhere in the contract:

- The currency is USD: <cat:LineExtensionTotalAmount currencyID="USD">1000</cat:LineExtensionTotalAmount>
- The seller's product code: <cat:SellersItemIdentification><cat:ID>32145-12</cat:ID></cat:SellersItemIdentification>

In addition, it contains room for other information, which isn't actually provided anywhere:

```

<cat:BuyerContact>
  <cat:ID/>
  <cat:Name/>
</cat:BuyerContact>

```

and buyer's address and deliver address.

Put another way, it is possible for the information in the XForms instance to be a superset of what appears in the structure. In other words, for there to be data in the instance which isn't pointed to by any <output> element.<sup>6</sup>

---

<sup>6</sup>And this doesn't disappear during xforms-model-construct or xforms-model-construct-done, since if you add a form submission element and submit the instance, this data is still there and submitted (tested with XSmiles 0.9 beta).

If the entire XForm is the eContract, then it is probable that this ought to have contractual effect. On the other hand, if the XForm is looked at in the browser, then basic principles (and XML Signatures) would suggest that anything like that which you can't see would not be part of the contract.

More discussion is required around this.

### **Discussion – date formatting**

In the instance, the date appeared as follows.

```
<cat:RequestedDeliveryDate>2004-06-30</cat:RequestedDeliveryDate>
```

It is quite possible that the vertical industry schema requires the date to appear in a specific format.

How do we ensure that when the contract is rendered, it appears in our preferred format ie 30 June 2004? Doing so means using XPath expressions, and possibly extension functions.

### **Example 3 – P3P**

Appendix one of this document contains an example based on the first example in section 3.1 of the Platform for Privacy Preferences 1.1 (P3P1.1) W3C Working Draft of 10 February 2004.

In that example, there seems to be an "impedance mismatch" between the P3P representation and the human readable policy in the "What We Collect and Why" section, and the "Data Retention" section.

There is nothing our XForms approach can do about that; its a fact of the data. It would be interesting to see whether IBM's EPAL is a better fit.

Nevertheless, the lesson is very clearly that this type of semantic markup is most useful if the instance contains the exact words which need to appear in the output (XPath expressions and extension functions notwithstanding).

Note from the appendix that one of the words we want to mark up semantically in this example is "correct", but that it appears in the P3P instance as an element name, rather than PCDATA or an attribute value. This is not a problem, since we can get the element name using XPath's local-name(), in either of two ways:

- a calculation which the <output> element points to:

```
<xforms:bind nodeset="/POLICIES/p3p:POLICY/p3p:DISPUTES-GROUP/p3p:DISPUTES/p3p:REMEDIES/p3p:correct" id="remedy" calculate="local-name()" />
```

- using @value, instead of @ref:

```
<xforms:output value="local-name(/POLICIES/p3p:POLICY/p3p:DISPUTES-GROUP/p3p:DISPUTES/p3p:REMEDIES/*)"/>
```

### **How does the semantic information get marked up?**

Somehow, the semantic information has to be marked up, if it is to be useful. In other words, bits of text in the structural markup need to be bound to values in the XForms instance.

I can see four use cases:

1. A contracts creation application creates a complete eContract (ie with the instance

populated and the bindings in place), all ready for some downstream application to use.

2. Similar to (1), an application merges instance data with a contract template, to create a complete eContract.
3. An end user is using a "marking pen" application to add semantic markup to a structural representation of the contract (ie <section>s and <p>s). They pick things from the vertical schema, and identify the words in the contract to bind it to.
4. An end user types values into the contract schedule using XForm form controls, and submits the form. This is using XForms technology in the way it was intended.

In use cases 1-3, the eContract doesn't actually have any visible form fields or a submit button.

## **Leveraging XForms**

As shown above, we can use the concepts introduced by XForms to bind a contract structure to vertical industry instance data. To do this, we only need a very small subset of what is in the XForms specification.

The 4 use cases identified above don't involve sending an XForm eContract to an XForm-enabled web browser.

Is this a useful thing to do? The answer is quite probably:

- to view it
- it is useful to be able to send an offer, which the user can accept by clicking

## **Example 4 – Click-through licence**

Here is a click through licence, which is the first example which has a submit button.

<http://java.sun.com/j2se/1.4.2/download.html>

Sun Microsystems, Inc. Binary Code License Agreement for the JAVA(TM) 2 SOFTWARE DEVELOPMENT KIT (J2SDK), STANDARD EDITION, VERSION 1.4.2\_X

SUN MICROSYSTEMS, INC. ("SUN") IS WILLING TO LICENSE THE SOFTWARE IDENTIFIED BELOW TO YOU ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE LICENSE AGREEMENT AND SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT"). PLEASE READ THE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE TERMS OF THE AGREEMENT. INDICATE ACCEPTANCE BY SELECTING THE "ACCEPT" BUTTON AT THE BOTTOM OF THE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY ALL THE TERMS, SELECT THE "DECLINE" BUTTON AT THE BOTTOM OF THE AGREEMENT AND THE DOWNLOAD OR INSTALL PROCESS WILL NOT CONTINUE.

:

2. LICENSE TO USE. Subject to the terms and conditions of this Agreement, including, but not limited to the Java Technology Restrictions of the Supplemental License Terms, Sun grants you a non-exclusive, non-transferable, limited license without license fees to reproduce and use internally Software complete and unmodified for the sole purpose of running Programs. Additional licenses for developers and/or publishers are granted in the Supplemental License Terms.

3. RESTRICTIONS. Software is confidential and copyrighted. Title to Software and all associated intellectual property rights is retained by Sun and/or its licensors. Unless enforcement is prohibited by applicable law, you may not modify, decompile, or reverse engineer Software. You acknowledge that Licensed Software is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Sun Microsystems, Inc. disclaims any express or implied warranty of fitness for such uses. No right, title or interest in or to any trademark, service mark, logo or trade name of Sun or

its licensors is granted under this Agreement. Additional restrictions for developers and/or publishers licenses are set forth in the Supplemental License Terms.

:

## ... using XrML2

I looked at using XrML2 to capture the semantics of this licence.

Although XrML contains the sorts of concepts you'd expect in order to describe rights, for example:

- identification of the thing in which the rights are being granted
- identification of the person granting the rights ("issuer")
- identification of the person to whom they are being granted ("principal")
- a description of the rights being granted
- the price

there is again an "impedance mismatch":

- the issuer is described by a digital signature, rather than name, address etc
- here we want to describe the rights "reproduce and use", but XrML2 is more specific, giving us verbs such as "backup", "copy", "delete", "execute", "install".

## ... using ODRL 1.1

Similarly ODRL 1.1 contains the sorts of concepts you'd expect, along with the "impedance mismatch":

- parties are identified by a URI
- the rights are again more fine grained than "reproduce and use"

Interestingly, ODRL has concepts of "offer" and "agreement", so the web page displaying the terms could use the offer XML, and assuming the user clicked "I agree", then agreement XML could be sent to them.

## ... so a thought experiment

Assume a suitable "software licence agreement markup language" (SLAM).

Assume an XForm eContract offer – containing SLAM and eContracts structural markup bound in an XForm with a submit button - was displayed in the web browser. The user clicks "I accept", and wants to save a copy of the agreement to their hard drive.

There are three different views of the contract they may wish to end up with:

- SLAM only: this is what `<xforms:submission id="SaveLocal" action="file://c:/software-licence.xml" method="put" />` would give you
- XForm: this is what "File > Save" would give you. This is useful for downstream eContracts contract management software, provided "<output>" is the only form widget used (any others would allow the user to enter new data which "File > Save" wouldn't capture. It isn't human readable without XForms aware software though.
- A human readable document eg PDF, RTF, XHTML (without XForms elements), SVG etc.

## **Example 5 – FpML**

In this example, I use FpML and XForms to markup the semantics in the following simplified/fictional Confirmation Notice:

F.R.A. CONTRACT  
CONFIRMATION NOTICE

TO:- Midland Bank

FROM:- ABN AMRO Bank

We are pleased to confirm the following Forward Rate Agreement ('F.R.A.') made between ourselves as per FICTIONAL Recommended Terms and Conditions

CONTRACT CURRENCY & AMOUNT: CHF 25,000,000

PAYMENT DATE:	<b>1991-07-17</b>
TERMINATION DATE:	<b>1992-01-17</b>
CONTRACT PERIOD (DAYS)	184
CONTRACT RATE	0.04, ACT/360
NON-STANDARD TERMS & CONDITIONS (IF ANY)	

PLEASE ADVISE BY TELEX, OR CABLE US IMMEDIATELY, SHOULD THE PARTICULARS OF THIS CONFIRMATION NOT BE IN ACCORDANCE WITH YOUR UNDERSTANDING.

The XForm is in appendix 2.

Again in the real world, there are probably date formats to worry about here, as well as numerical conversions (eg 0.04 to 4%).

### **Why not just use XSLT?**

When the document is so short, it is natural to ask why you wouldn't just apply XSLT to the FpML to create your confirmation.

The answer is that you could.

But the document which is required for business purposes is the confirmation. So that's the document which should be archived etc. We'd like it to be an XML document, and we'd like that XML to capture the semantics – which is a requirement that the XForms approach addresses – as well as the rest of the human readable text of the confirmation. So XForms is

a good approach.

You may well still use XSLT to create an XForms based confirmation from the FpML. XForms processing software can then manipulate the XForm as required in the business process.

## ***Legal Semantic Markup***

I proposed the following requirements:

- attach semantic markup to phrases
- attach semantic markup to clauses
- attach semantic markup to groups of clauses

What sort of things might a lawyer wish to capture in semantic markup?

- jurisdiction
- governing law
- term and renewal details, including rights to terminate on notice
- clauses which survive termination
- effects of change in control
- obligations
- warranties

Some of these things occur once in the contract and are relatively simple to mark up (for example, jurisdiction and governing law); other occur once but would be quite complex to parameterize in a schema language (eg term and renewal details); others occur repeatedly throughout the contract.

The XForms <output> element is appropriate for the simple stuff.<sup>7</sup>

Assuming there is a compelling business case for capturing the other stuff (how much of it would you want to exchange with the other party?), how could we do it? Does it fit into the XForms framework explored in this paper?

The idea of storing the semantic markup in an XForms instance element still works. The markup would use a "Contract Semantic Markup Language" which would need to be developed by the eContracts TC.

The legal semantic markup would be tied to the structural markup, but it wouldn't do so using XForms output elements.

Conceptually, either the semantic markup could point to the structural (using XPath expressions or just IDs) – which seems natural since the legal semantic markup is "about" the structural, and shouldn't need to alter it - or the structural could point to the semantic (which is what XForms does).

Either would work fine, and wouldn't cause any problems for generic XForms processing

---

<sup>7</sup>Note: you can only apply one <output> to a given bit of text, since <output> can't be nested or overlap . An interesting case is if the schema provides for a delivery address and an invoice address, but no mechanism for saying they are the same. You'd have to decide whether the address appearing in the structure should be linked to delivery, or invoice address.

software. However, support for the chosen mechanism would need to be specifically built into eContracts software where relevant.

### ***Multiple vertical industry schemas***

The beauty of XForms is that it allows multiple semantic markup languages to co-exist.

For example, one eContract could simultaneously contain UBL semantic markup, and our own Contract Semantic Markup Language.

In this way, if the vertical industry schema does not have a suitable element for capturing a party's legal name and/or abbreviated name, we can capture that in a second model or instance.

### ***Outreach***

Some vertical schemas appear to map much more closely to phrases found in traditional contracts than others.

The ones which map closely are more suited to an XForms approach.

More work needs to be done to identify vertical industries which could benefit from eContracts semantic markup. I think we need to work closely with their standards bodies to verify their need/interest, and then test this approach on their contracts.

### ***Conclusions***

1. The XForms binding concept and its <output> element allows us to associate eContracts structure with existing vertical industry schemas
2. This simply works if the words of the contract which we wish to markup appear as PCDATA in the vertical industry schema
3. XForms is a natural approach, since we're using XHTML2 for structure.
4. If the data (eg date) is stored in the vertical industry schema in another format, then a mechanism for reformatting it is required.
5. Sometimes there is an "impedance mismatch" between the vertical industry schema and the text of the contract. This means that that particular vertical industry schema does not capture the semantics of the contract in a way a lawyer would expect and/or the contract doesn't contain all the terms the vertical industry schema anticipates.
6. The legal significance of the data contained in the vertical industry schema needs to be discussed – particularly where it isn't reflected in the text of the contract.
7. The eContracts TC should contact various of the vertical industry standards groups to establish their interest in exploring this approach further.
8. Legal semantic markup can be distinguished from the data vertical industry schemas typically capture. The XForms approach can be extended to accommodate legal semantic markup in a natural way.
9. XForms allows multiple semantic schemas to co-exist in the one document. This is likely to be useful, particularly if the eContracts TC creates a Contract Semantic Markup Schema to augment the existing vertical industry schemas.

## Appendix One – P3P

This example is based on the first example in section 3.1 of the Platform for Privacy Preferences 1.1 (P3P1.1) W3C Working Draft of 10 February 2004.

The English language policy is:

At CatalogExample, we care about your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it with information we could use to identify you.

CatalogExample, Inc. is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by holding Web site licensees to high privacy standards and confirming with independent auditors that these information practices are being followed.

Questions regarding this statement should be directed to:

CatalogExample  
4000 Lincoln Ave.  
Birmingham, MI 48009 USA  
email: catalog@example.com  
Telephone 248-EXAMPLE (248-392-6753)

If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample at <http://www.privacyseal.example.org>. CatalogExample will correct all errors or wrongful actions arising in connection with the privacy policy.

### *What We Collect and Why:*

When you browse through our site we collect:

- the basic information about your computer and connection to make sure that we can get you the proper information and for security purposes.
- aggregate information on what pages consumers access or visit to improve our site.

### *Data retention:*

We purge every two weeks the browsing information that we collect.

Using eContracts structural markup, this might look like:

```
<html>
  <head>
    <title>P3P Policy</title>
  </head>
  <body>
    <section>
      <section>
        <p>
```

At CatalogExample, we care about your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it with information we could use to identify you.

```
</p>
```

```

    </section>
  <section>
    <p>
CatalogExample, Inc. is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by
holding Web site licensees to high privacy standards and confirming with independent auditors that these information practices are
being followed.
    </p>
  </section>
  <section>
    <p>
Questions regarding this statement should be directed to:
CatalogExample
4000 Lincoln Ave.
Birmingham, MI 48009 USA
email: catalog@example.com
Telephone 248-EXAMPLE (248-392-6753)
    </p>
  </section>
  <section>
    <p>
If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample
at http://www.privacyseal.example.org. CatalogExample will correct all errors or wrongful actions arising in connection with the
privacy policy.
    </p>
  </section>
  <section>
    <h>What We Collect and Why</h>
    <p>
When you browse through our site we collect:
    </p>
    <section>
      <p>
the basic information about your computer and connection to make sure that we can get you the proper information and for security
purposes.
      </p>
    </section>
    <section>
      <p>
aggregate information on what pages consumers access or visit to improve our site.
      </p>
    </section>
    <section>
      <h>Data retention</h>
      <p>
We purge every two weeks the browsing information that we collect.
      </p>
    </section>
  </section>
</body>
</html>

```

Creating an XForm by merging that with the XML encoding of the policy given in section 3.1.2 of the specification, we get:

```

<html xmlns="http://www.w3.org/1999/xhtml" xmlns:xforms="http://www.w3.org/2002/xforms" xmlns:ev="http://www.w3.org/2001/xml-
events" xmlns:p3p="http://www.w3.org/2002/01/P3Pv1">
  <head>
    <title>Privacy Policy</title>
    <xforms:model>
      <xforms:instance xmlns="">
        <POLICIES xmlns:p3p="http://www.w3.org/2002/01/P3Pv1">
          <p3p:POLICY p3p:name="forBrowsers"
p3p:discuri="http://www.catalog.example.com/PrivacyPracticeBrowsing.html" xml:lang="en">
            <p3p:ENTITY>
              <p3p:DATA-GROUP>
                <p3p:DATA p3p:ref="#business.name">CatalogExample</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.postal.street">4000 Lincoln Ave.</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.postal.city">Birmingham</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.postal.stateprov">MI</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.postal.postalcode">48009</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.postal.country">USA</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.online.email">catalog@example.com</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.telecom.telephone.intcode">1</p3p:DATA>
                <p3p:DATA p3p:ref="#business.contact-info.telecom.telephone.loccode">248</p3p:DATA>
              </p3p:DATA-GROUP>
            </p3p:ENTITY>
          </p3p:POLICY>
        </POLICIES>
      </xforms:instance>
    </xforms:model>
  </head>
  <body>
    <p>
    </p>
  </body>
</html>

```

```

        <p3p:DATA p3p:ref="#business.contact-info.telecom.telephone.number">3926753</p3p:DATA>
    </p3p:DATA-GROUP>
</p3p:ENTITY>
<p3p:ACCESS>
    <p3p:nonident/>
</p3p:ACCESS>
<p3p:DISPUTES-GROUP>
    <p3p:DISPUTES p3p:resolution-type="independent" p3p:service="http://www.PrivacySeal.example.org"
p3p:short-description="PrivacySeal.example.org">
        <IMG src="http://www.PrivacySeal.example.org/Logo.gif" alt="PrivacySeal's logo"/>
    <p3p:REMEDIES>
        <p3p:correct/>
    </p3p:REMEDIES>
    </p3p:DISPUTES>
</p3p:DISPUTES-GROUP>
<p3p:STATEMENT>
    <p3p:PURPOSE>
        <p3p:admin/>
        <p3p:develop/>
    </p3p:PURPOSE>
    <p3p:RECIPIENT>
        <p3p:ours/>
    </p3p:RECIPIENT>
    <p3p:RETENTION>
        <p3p:stated-purpose/>
    </p3p:RETENTION>
    <p3p:DATA-GROUP>
        <p3p:DATA p3p:ref="#dynamic.clickstream"/>
        <p3p:DATA p3p:ref="#dynamic.http"/>
    </p3p:DATA-GROUP>
</p3p:STATEMENT>
</p3p:POLICY>
</POLICIES>
</xforms:instance>
<xforms:bind nodeset="/POLICIES/p3p:POLICY/p3p:DISPUTES-GROUP/p3p:DISPUTES/p3p:REMEDIES/p3p:correct"
id="remedy" calculate="local-name()"/>
</xforms:model>
</head>
<body>
    <section>
        <section>
            <p>
                At <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.name]"/>, we care about
                your privacy. When you come to our site to look for an item, we will only use this information to improve our site and will not store it
                with information we could use to identify you.
            </p>
        </section>
        <section>
            <p>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.name]"/>,
                Inc. is a licensee of the PrivacySealExample Program. The PrivacySealExample Program ensures your privacy by holding Web site
                licensees to high privacy standards and confirming with independent auditors that these information practices are being followed.
            </p>
        </section>
        <section>
            <p>
                Questions regarding this statement should be directed to:<br/>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.name]"/>
                <br/>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.contact-
                info.postal.street]"/>
                <br/>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.contact-
                info.postal.city]"/>,
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.contact-info.postal.stateprov]"/>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.contact-
                info.postal.postalcode]"/>
                <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref="#business.contact-
                info.postal.country]"/>
                <br/>
            </p>
        </section>
    </section>
</body>

```

email: <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref='#business.contact-info.online.email']"/>  
Telephone <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref='#business.contact-info.telecom.telephone.loccode']"/>-EXAMPLE (<xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref='#business.contact-info.telecom.telephone.loccode']"/>-<xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref='#business.contact-info.telecom.telephone.number']"/>)

</p>  
</section>  
<section>  
<p>

If we have not responded to your inquiry or your inquiry has not been satisfactorily addressed, you can contact PrivacySealExample at <xforms:output ref="p3p:POLICY/p3p:DISPUTES-GROUP/p3p:DISPUTES/@p3p:service"/>. <xforms:output ref="p3p:POLICY/p3p:ENTITY/p3p:DATA-GROUP/p3p:DATA[@p3p:ref='#business.name']"/> will <xforms:output bind="remedy"/> all errors or wrongful actions arising in connection with the privacy policy.

</p>  
</section>  
<section>  
<h>What We Collect and Why</h>  
<p>

When you browse through our site we collect:<p>

<section>

the basic information about your computer and connection to make sure that we can get you the proper information and for security purposes.

</section>  
<section>

aggregate information on what pages consumers access or visit to improve our site. </section>

</section>  
<section>  
<h>Data retention</h>  
<p>

We purge every two weeks the browsing information that we collect.

</p>  
</section>  
</section>

</body>  
</html>

## Appendix Two – FpML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:xforms="http://www.w3.org/2002/xforms" xmlns:ev="http://www.w3.org/2001/xml-
events">
  <head>
    <title>Forward Rate Agreement - Confirmation</title>
    <link rel="stylesheet" href="/main.css" type="text/css"/>
    <xforms:model>
      <xforms:instance xmlns="">
        <FpML version="1-0"
          businessCenterSchemeDefault="http://www.fpml.org/spec/2000/business-center-1-0"
          businessDayConventionSchemeDefault="http://www.fpml.org/spec/2000/business-day-
convention-1-0"
          currencySchemeDefault="http://www.fpml.org/ext/iso4217"
          dateRelativeToSchemeDefault="http://www.fpml.org/spec/2001/date-relative-to-1-0"
          dayCountFractionSchemeDefault="http://www.fpml.org/spec/2000/day-count-fraction-1-0"
          dayTypeSchemeDefault="http://www.fpml.org/spec/2000/day-type-1-0"
          floatingRateIndexSchemeDefault="http://www.fpml.org/ext/isda-1991-definitions"
          partyIdSchemeDefault="http://www.fpml.org/ext/iso9362"
          periodSchemeDefault="http://www.fpml.org/spec/2000/period-1-0">
          <trade>
            <tradeHeader>
              <partyTradeIdentifier>
                <partyReference href="#MIDLAND"/>
                <tradeId tradeIdScheme="http://www.hsbc.com/swaps/trade-id">MB87623</tradeId>
              </partyTradeIdentifier>
              <partyTradeIdentifier>
                <partyReference href="#ABNAMRO"/>
                <tradeId tradeIdScheme="http://www.abnamro.com/swaps/trade-id">AA9876</tradeId>
              </partyTradeIdentifier>
              <tradeDate>1991-05-14</tradeDate>
            </tradeHeader>
            <product>
              <fra>
                <buyerPartyReference href="#MIDLAND"/>
                <sellerPartyReference href="#ABNAMRO"/>
                <adjustedEffectiveDate id="resetDate">1991-07-17</adjustedEffectiveDate>
                <adjustedTerminationDate>1992-01-17</adjustedTerminationDate>
                <paymentDate>
                  <unadjustedDate>1991-07-17</unadjustedDate>
                  <dateAdjustments>
                    <businessDayConvention>FOLLOWING</businessDayConvention>
                    <businessCenters>
                      <businessCenter>CHZU</businessCenter>
                    </businessCenters>
                  </dateAdjustments>
                </paymentDate>
                <fixingDateOffset>
                  <periodMultiplier>-2</periodMultiplier>
                  <period>D</period>
                  <dayType>Business</dayType>
                  <businessDayConvention>NONE</businessDayConvention>
                  <businessCenters>
                    <businessCenter>GBLO</businessCenter>
                  </businessCenters>
                  <dateRelativeTo href="#resetDate">ResetDate</dateRelativeTo>
                </fixingDateOffset>
                <dayCountFraction>ACT/360</dayCountFraction>
                <calculationPeriodNumberOfDays>184</calculationPeriodNumberOfDays>
                <notional>
                  <currency>CHF</currency>
                  <amount>25000000.00</amount>
                </notional>
                <fixedRate>0.04</fixedRate>
                <floatingRateIndex>CHF-LIBOR-BBA</floatingRateIndex>
                <indexTenor>
                  <periodMultiplier>6</periodMultiplier>
                  <period>M</period>
                </indexTenor>
              </fra>
            </product>
          </trade>
        </FpML>
      </xforms:instance>
    </xforms:model>
  </head>
</html>
```

```

        <fraDiscounting>true</fraDiscounting>
    </fra>
</product>
<party id="MIDLAND">
    <partyId>MIDLGB22</partyId>
</party>
<party id="ABNAMRO">
    <partyId>ABNANL2A</partyId>
</party>
</trade>
</FpML>
</xforms:instance>
</xforms:model>
</head>
<body>
<h>F.R.A. CONTRACT</h>
<h>CONFIRMATION NOTICE</h>
<p>TO:- Midland Bank</p>
<p>FROM:- ABN AMRO Bank</p>
<p>We are pleased to confirm the following Forward Rate Agreement ('F.R.A.') made between ourselves as per FICTIONAL
Recommended Terms and Conditions </p>
<table>
    <tr>
        <td>CONTRACT CURRENCY & AMOUNT:</td>
        <td>
            <xforms:output ref="trade/product/fra/notional/currency"/>
            <xforms:output ref="trade/product/fra/notional/amount"/>
        </td>
    </tr>
    <tr>
        <td>PAYMENT DATE:</td>
        <td>
            <xforms:output ref="trade/product/fra/paymentDate/unadjustedDate"/>
        </td>
    </tr>
    <tr>
        <td>TERMINATION DATE:</td>
        <td>
            <xforms:output ref="trade/product/fra/adjustedTerminationDate"/>
        </td>
    </tr>
    <tr>
        <td>CONTRACT PERIOD (DAYS)</td>
        <td>
            <xforms:output ref="trade/product/fra/calculationPeriodNumberOfDays"/>
        </td>
    </tr>
    <tr>
        <td>CONTRACT RATE</td>
        <td>
            <xforms:output ref="trade/product/fra/fixRate"/>, <xforms:output ref="trade/product/fra/dayCountFraction"/>
        </td>
    </tr>
    <tr>
        <td>NON-STANDARD TERMS & CONDITIONS (IF ANY)</td>
        <td></td>
    </tr>
</table>
<p>PLEASE ADVISE BY TELEX, OR CABLE US IMMEDIATELY, SHOULD THE PARTICULARS OF THIS
CONFIRMATION NOT BE IN ACCORDANCE WITH YOUR UNDERSTANDING.</p>
</body>
</html>

```