XHTML 2 for LegalXML – The eContracts Experience

Presentation to OASIS LegalXML Member Section
28 April 2004

Jason Harrop

SPEEDLEGAL

leaders in enterprise document automation

eContracts at OASIS

- Participant profiles
  - 3-4 (mainly small) software vendors / consultants
  - 4-5 lawyers (USA, Australian)
  - 1-2 academics
  - and others
- .. interested in:
  - Authoring of contracts
  - .. or “business documents” more generally?
  - Negotiation of contracts
    - Human to human
    - Computer
  - Contract management
  - Particular technical solutions
  - “Formal modeling”

SPEEDLEGAL

(ctd)

Mission

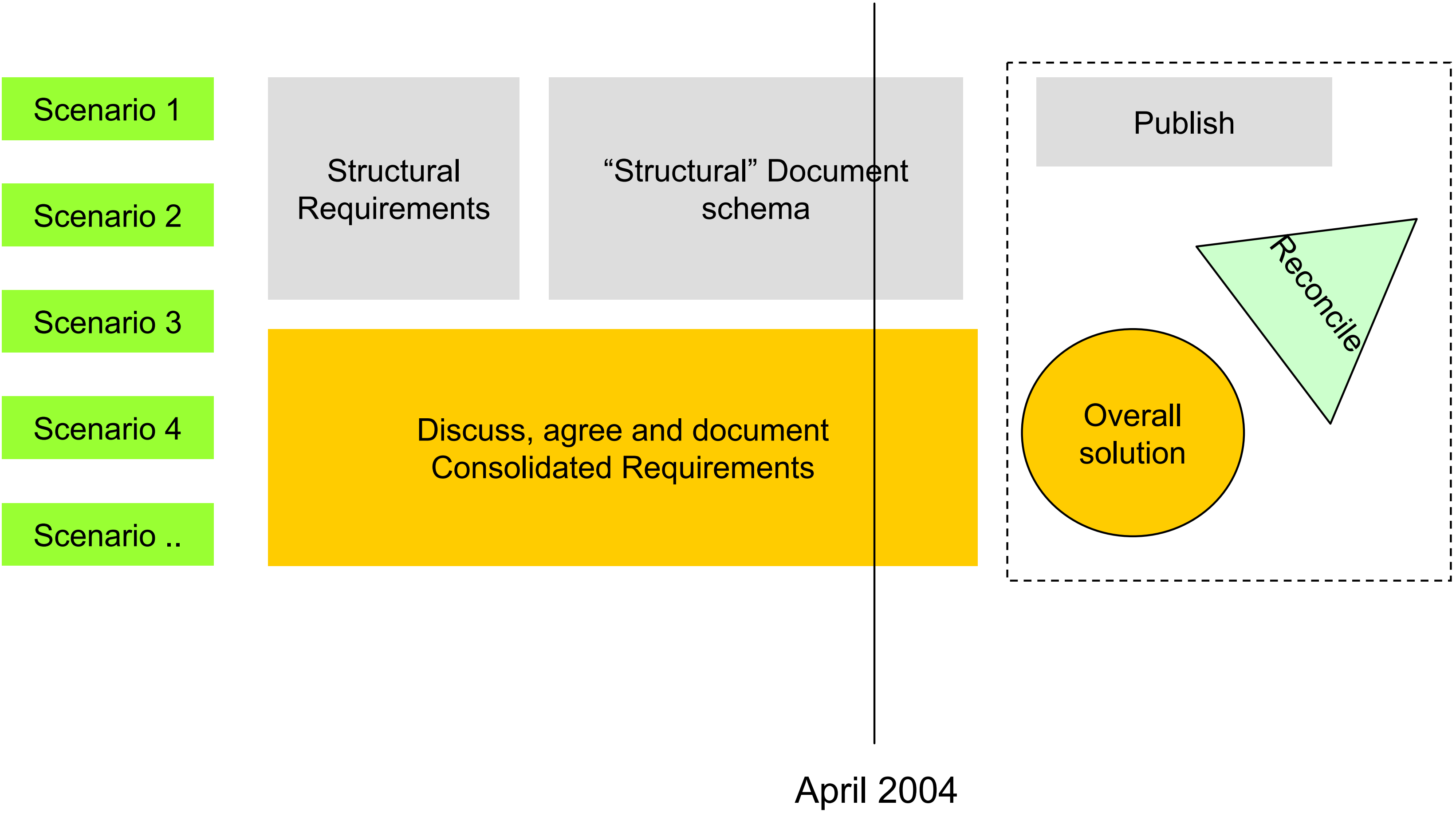
The eContracts Technical Committee exists to develop open XML standards for the markup of contract documents to enable the efficient creation, maintenance, management, exchange and publication of contract documents and contract terms.

Scope

(i). The core scope of this activity will be to the creation of DTD(s) / schema(s) that can be used by parties:

a. Negotiating and finalizing contracts in an application neutral format;
b. Exchanging contract contents as valid XML;
c. Automating processing of contract content, for example for use in contract management applications;
d. To support the production of human readable output documents; and
e. To facilitate the use of reusable or boilerplate information within a contract.

Actual process


Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario ..

Structural Requirements

"Structural" Document schema

Discuss, agree and document Consolidated Requirements

Publish

Reconcile

Overall solution

April 2004

SPEEDLEGAL

Structure – “Clause Patterns”

Example 1A: A clause with a heading

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Example 1B: A clause with a heading

Article 1 - Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Example 1C: .. with multiple paragraphs

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Example 2: A clause without a heading

1. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Structure – “Clause Patterns” (ctd)

Example 3A: A clause with subclauses with headings

1. Heading

1.1 Sub-heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

1.2 Sub-heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Example 3B: .. with subclauses without headings

1. Heading

1.1 Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

1.2 Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

SPEEDLEGAL

Structure – “Clause Patterns” (ctd)

Example 4A: A clause with a block list

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. ***My list includes:***

(a) apples; and

(b) oranges,

but excludes pears. More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Example 4B: A clause with an inline list

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. ***My list includes: (a) apples; and (b) oranges, but excludes pears.*** More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

Structure – combining the patterns

1. Leased Premises; Term of Lease

1.1 Leased Premises

Landlord leases to Tenant, and Tenant rents from Landlord, subject to the Permitted Encumbrances, the land located in the City, County and State of New York more particularly described on Exhibit A hereto (the "Land"),

TOGETHER WITH (i) all Improvements (excluding any thereof which pursuant to the final paragraph of this Section 1.1 are not Landlord's property), and (ii) all personal property owned by Landlord now or hereafter attached to or used in connection with the Improvements,

TOGETHER WITH all right, title and interest, if any, of Landlord in and to:

(a) any strips and gores of land adjoining the Land on any side thereof;

(b) any land lying in the bed of any street or avenue abutting the Land, to the center line thereof; and

(c) any easements or other rights in adjoining property enuring to Landlord by reason of ownership of the Land;

all of the foregoing (together with any Improvements excluded from clause (i) above) are collectively called the "Leased Premises".

1.2 Term

Except as otherwise provided in Section 1.3, the term of this Lease….

Structure – the “List – sub-clause continuum”

Even experienced authors draw the line in different places!

2003

- Business requirements include
  - formal “contract document”
  - simplicity of authoring

- Vote

2003 (ctd)

- Review of existing DTDs
    - XHTML 1, DocBook, OpenOffice etc
    - None suitable

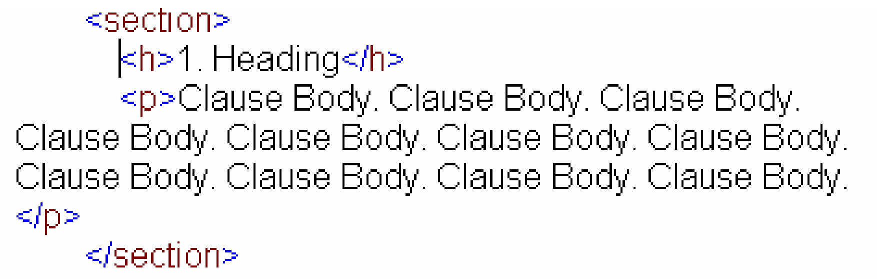
- Solution – attempt 1
    - Jason Harrop and Peter Meyer
    - a lot of work done
    - .. culminating in joint presentation at Oasis Open Standards Conference in Sydney
    - but still some contentious issues
    - .. which we couldn’t resolve in time to present a joint report back to the TC before the deadline imposed on us

XHTML2 ? (ctd)

Example 1A: A clause with a heading

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.



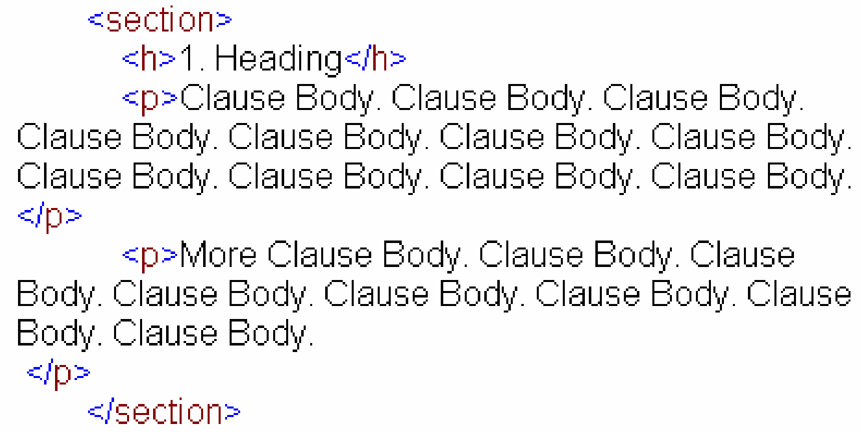
```
        <section>
          <h>1. Heading</h>
          <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
        </section>
```

Example 1C: .. with multiple paragraphs

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.



```
        <section>
          <h>1. Heading</h>
          <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
          <p>More Clause Body. Clause Body. Clause
Body. Clause Body. Clause Body. Clause Body. Clause
Body. Clause Body.
 </p>
        </section>
```

SPEEDLEGAL

XHTML2 ? (ctd)

Example 3A: A clause with subclauses with headings

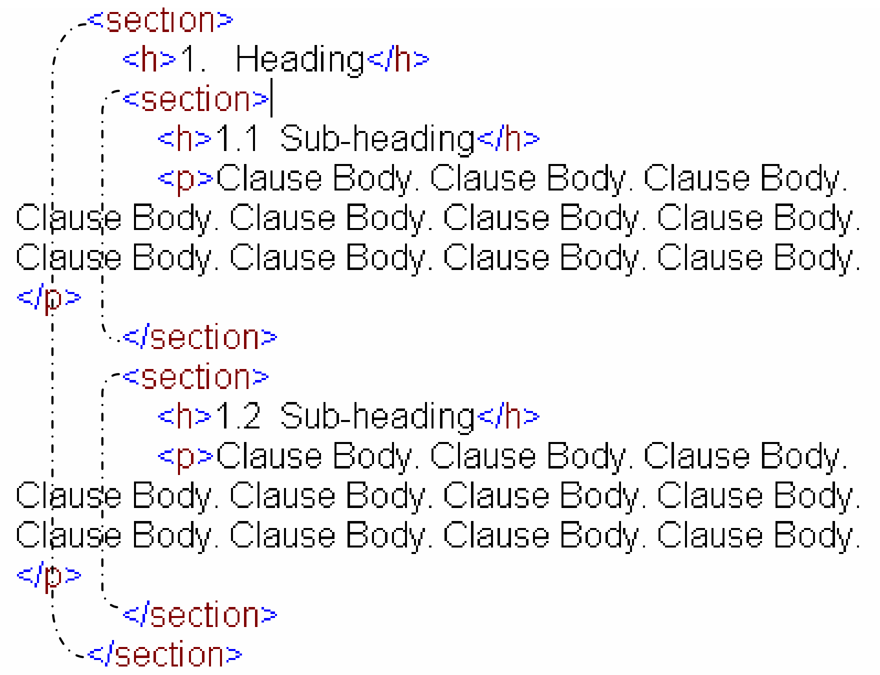
1. Heading

1.1 Sub-heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

1.2 Sub-heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.



```
<section>
    <h>1.   Heading</h>
    <section>
        <h>1.1  Sub-heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
    </section>
    <section>
        <h>1.2  Sub-heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
    </section>
</section>
```

XHTML2 ? (ctd)

Example 4A: A clause with a block list

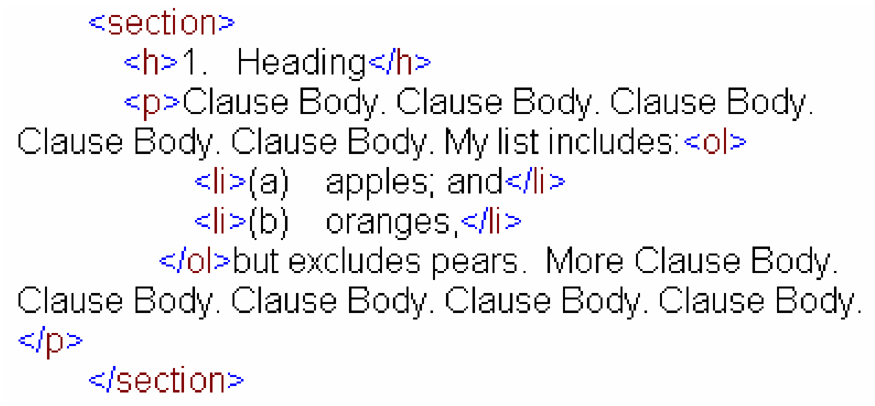
1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. ***My list includes:***

(a) apples; and

(b) oranges,

but excludes pears. More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.



```
     <section>
        <h>1.   Heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. My list includes:<ol>
              <li>(a)    apples; and</li>
              <li>(b)    oranges,</li>
           </ol>but excludes pears.  More Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
     </section>
```

XHTML2 for structure – why XHTML 2 ?

0. .. So it looks like it has the potential to do the job

1. Display in web browsers, even if there is no stylesheet

2. Assist with market acceptance of TC work

3. Willingness to learn XHTML2 amongst developers (but no familiarity, much less amongst contract managers or lawyers).

4. XHTML2 includes an XForms module

5. Content reuse from other XHTML 2.0 documents

6. Editors designed for strictly conforming XHMTL 2.0 documents and/or Host Language documents?

.. but !! [Jan 2004]

1. Many irrelevant and inappropriate elements and attributes make editing daunting and confusing for lawyers and contract managers, and processing unnecessarily complex for a variety of applications

- Specification allows #PCDATA in <section> element
- Strange content model for <li> element (#PCDATA, or <p> or <section>)
- Lists in <section> element (ie not just in <p>)
- Irrelevant block content (eg <blockcode>, <pre>, <address>)
- Irrelevant elements in the %Inline; content set (eg <code>, <kbd>, <var>)
- Inapplicable attributes
- Multiplicity of list types (<ol>, <dl>, <nl>, <ul>)

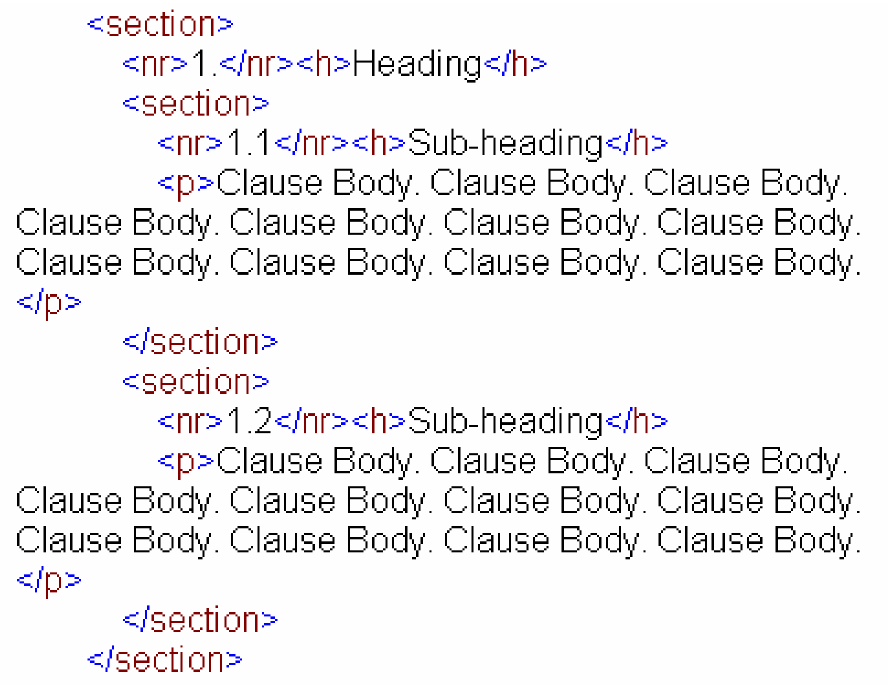
2. Structure module might need extension

http://lists.oasis-open.org/archives/legalxml-econtracts/200401/msg00022.html

So “prune” it - Structural Sub-committee [4 Feb 2004]

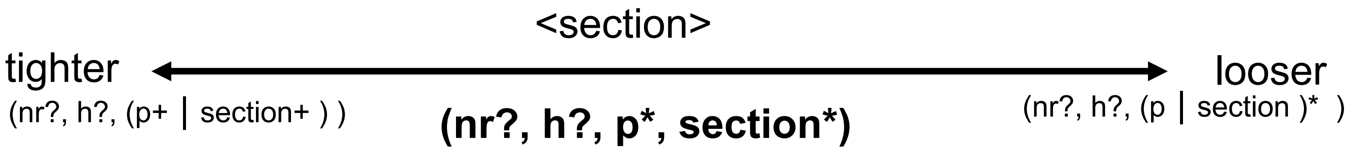
“the tc shall use xhtml 2 as the basis for our structural markup specification, deleting and adding to it as necessary”

Structural: Clause model



```
    <section>
      <nr>1.</nr><h>Heading</h>
      <section>
        <nr>1.1</nr><h>Sub-heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
      </section>
      <section>
        <nr>1.2</nr><h>Sub-heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
      </section>
    </section>
```

- Use <section>
- Tighten model:
  - Don’t allow #PCDATA directly in <section>
  - Don’t allow lists in <section>
- Invent <nr> element for clause number



<section>

tighter

(nr?, h?, (p+ | section+))

(nr?, h?, p*, section*)

looser

(nr?, h?, (p | section)*)

Structural: Paragraphs and lists

- “grammatical” paragraph model, so lists and sub-clauses are marked up differently, which means author must decide
- optional <l> for a line of text which needs to appear on a separate line
- discarded <code>, <kbd>, <samp>, <var>
- kept <abbr>, <sub>, <sup>, <cite>, <dfn>, <em>, <quote>, <span>, <strong>

- <ol> list element only for block lists – remove the others
- tighter content model for <li>: block list items contain headings and paragraphs, not #PCDATA directly

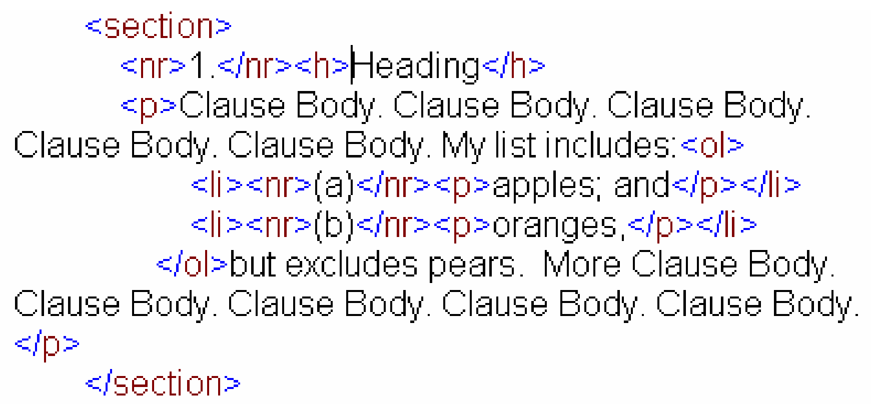
Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. My list includes:

(a) apples; and

(b) oranges,

but excludes pears. More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

One “grammatical” paragraph, not 4 word processor (or “simple”) paragraphs



```
        <section>
            <nr>1.</nr><h>Heading</h>
            <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. My list includes:<ol>
                  <li><nr>(a)</nr><p>apples; and</p></li>
                  <li><nr>(b)</nr><p>oranges,</p></li>
               </ol>but excludes pears.  More Clause Body.
Clause Body. Clause Body. Clause Body. Clause Body.
</p>
        </section>
```

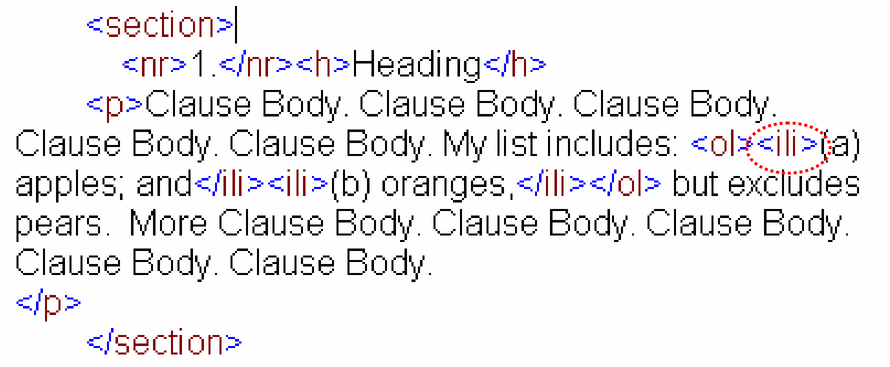
Structure – inline lists…

Example 4B: A clause with an inline list

1. Heading

Clause Body. Clause Body. Clause Body. Clause Body. Clause Body. ***My list includes: (a) apples; and (b) oranges, but excludes pears.*** More Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.

- Invent "inline list item" element, so that we can restrict its content to inline elements (ie no <p> or other block stuff)



```
        <section>
          <nr>1.</nr><h>Heading</h>
        <p>Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body. My list includes: <ol><ili>(a)
apples; and</ili><ili>(b) oranges,</ili></ol> but excludes
pears.  More Clause Body. Clause Body. Clause Body.
Clause Body. Clause Body.
</p>
        </section>
```

Structural: Higher level structure

Date
Parties
Recitals/Background
Operative Clauses
Signatures
Schedules/Annexures

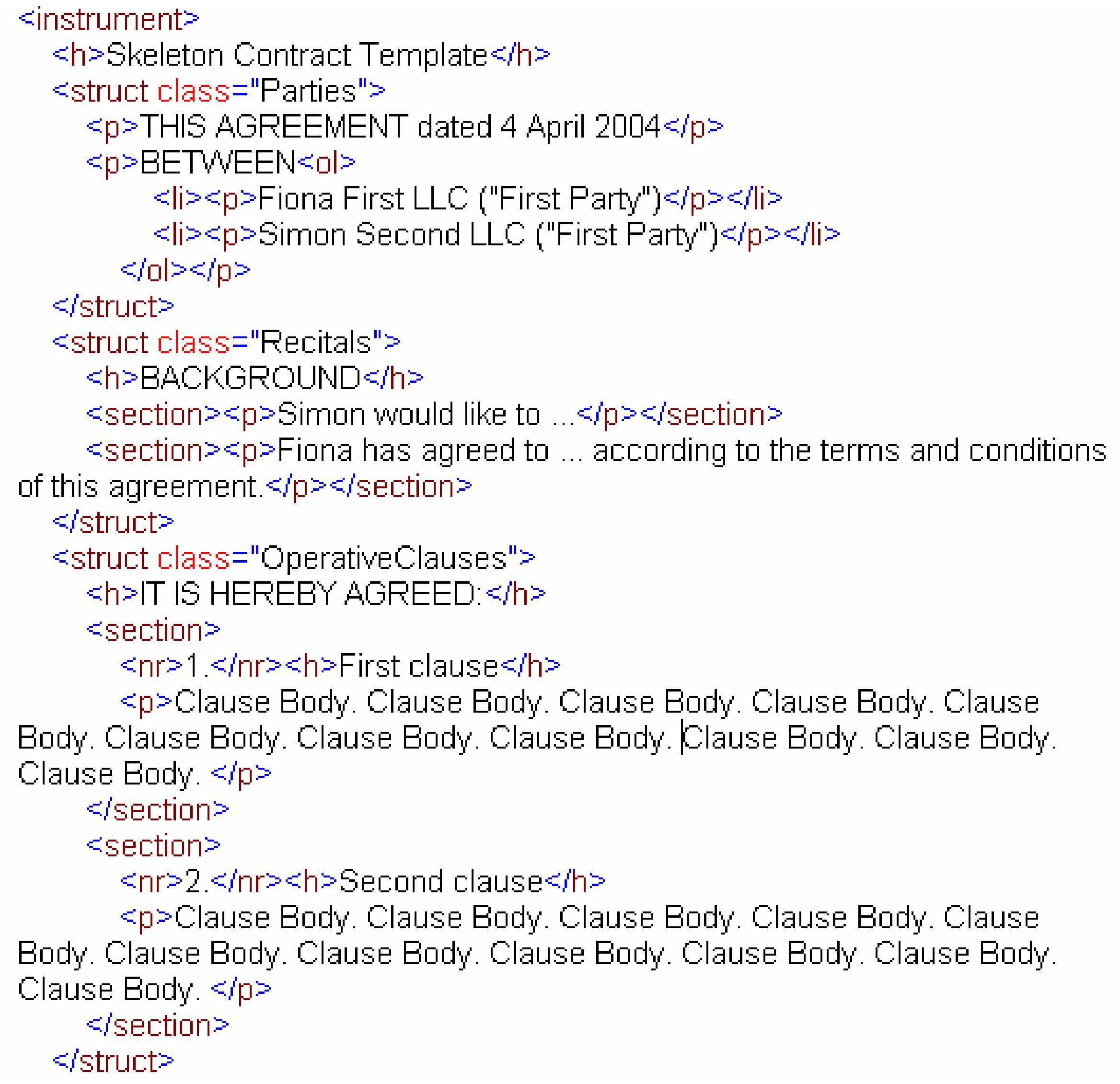
Common, but some other structure is equally possible.

- Invent an element “instrument”
  - Useful to contain documents included in schedules (eg Pro-Forma NDA) – these might not be contracts
  - Which can be re-used as the root element of the eContract
  - Feedback will be solicited from Member Section
- Invent a single recursive generic container called “struct” for now (name will be chosen soon)

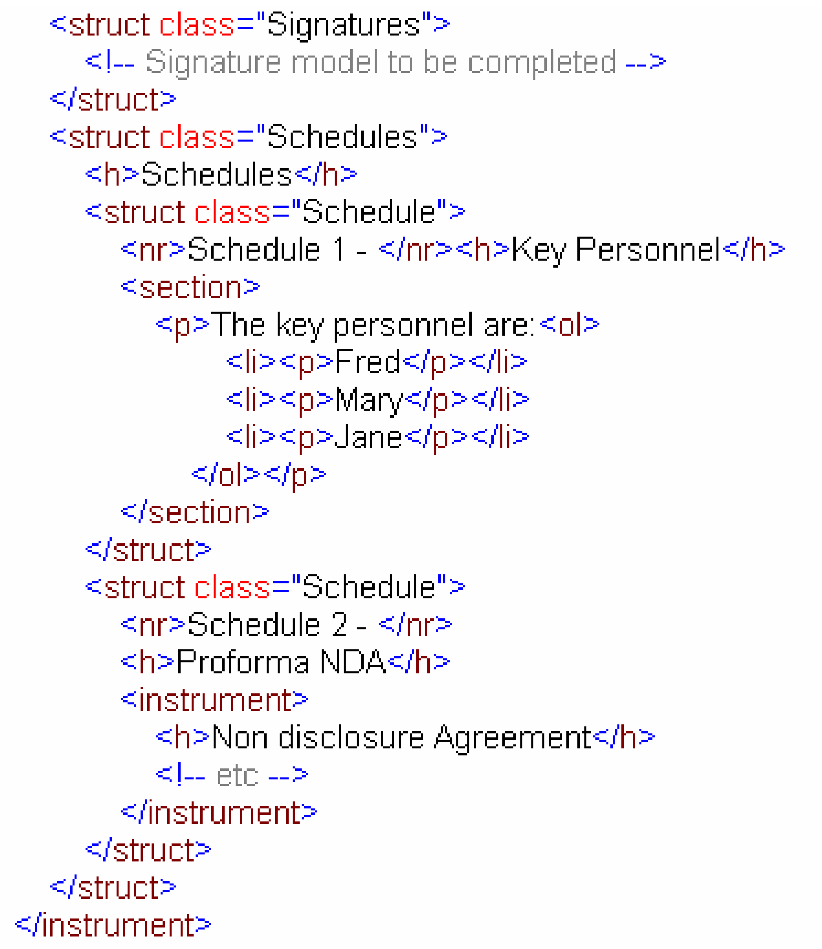
```
<!ELEMENT instrument (h*, struct*)>

<!ELEMENT struct (nr?, h?, p*, (section* | struct* | instrument*))>
```

SPEEDLEGAL

Putting it all together...



```
<instrument>
  <h>Skeleton Contract Template</h>
  <struct class="Parties">
    <p>THIS AGREEMENT dated 4 April 2004</p>
    <p>BETWEEN<ol>
        <li><p>Fiona First LLC ("First Party")</p></li>
        <li><p>Simon Second LLC ("First Party")</p></li>
      </ol></p>
  </struct>
  <struct class="Recitals">
    <h>BACKGROUND</h>
    <section><p>Simon would like to ...</p></section>
    <section><p>Fiona has agreed to ... according to the terms and conditions
of this agreement.</p></section>
  </struct>
  <struct class="OperativeClauses">
    <h>IT IS HEREBY AGREED:</h>
    <section>
      <nr>1.</nr><h>First clause</h>
      <p>Clause Body. Clause Body. Clause Body. Clause Body. Clause
Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. </p>
    </section>
    <section>
      <nr>2.</nr><h>Second clause</h>
      <p>Clause Body. Clause Body. Clause Body. Clause Body. Clause
Body. Clause Body. Clause Body. Clause Body. Clause Body. Clause Body.
Clause Body. </p>
    </section>
  </struct>
```



```
  <struct class="Signatures">
    <!-- Signature model to be completed -->
  </struct>
  <struct class="Schedules">
    <h>Schedules</h>
    <struct class="Schedule">
      <nr>Schedule 1 - </nr><h>Key Personnel</h>
      <section>
        <p>The key personnel are:<ol>
            <li><p>Fred</p></li>
            <li><p>Mary</p></li>
            <li><p>Jane</p></li>
          </ol></p>
      </section>
    </struct>
    <struct class="Schedule">
      <nr>Schedule 2 - </nr>
      <h>Proforma NDA</h>
      <instrument>
        <h>Non disclosure Agreement</h>
        <!-- etc -->
      </instrument>
    </struct>
  </struct>
</instrument>
```

SPEEDLEGAL

eContracts structure – what next?

- Finish 7 April discussion of higher level structure
  - Agree element name for “struct”
  - Values for @class
- Signature blocks
- Numbering, tables, styles
- Etc…

eContracts structure – some remaining challenges

- Tables – should we use CSS to do borders etc?
- @style
- What is the role of <div>?
- Cross references
- Page breaks, cover page, headers/footers?

Thoughts and Lessons so far

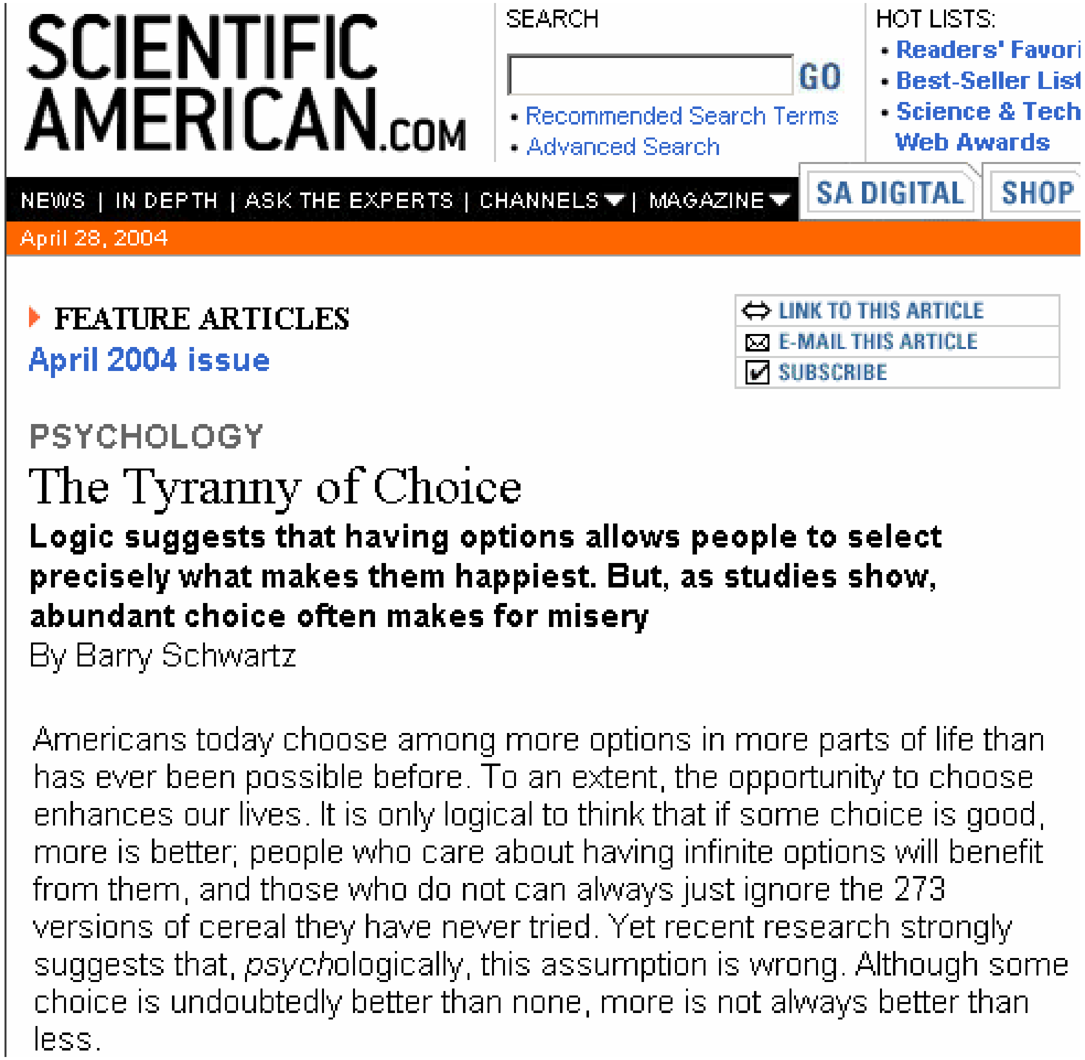
- Be as clear on requirements as you can be (but no clearer?)
  - Tension between generic solution, and solution to particular requirements

- Process matters – a lot!

- Don’t rush it (!)

- More than one solution: Ask “can you live with it?”
  - ***Maximizers*** "always aim to make the best possible choice“
  - ***Satisficers*** are "those who aim for 'good enough'.“

  An agreed framework (ie XHTML) helps a lot, because it provides helpful constraints on choice



PSYCHOLOGY

The Tyranny of Choice

Logic suggests that having options allows people to select precisely what makes them happiest. But, as studies show, abundant choice often makes for misery

By Barry Schwartz

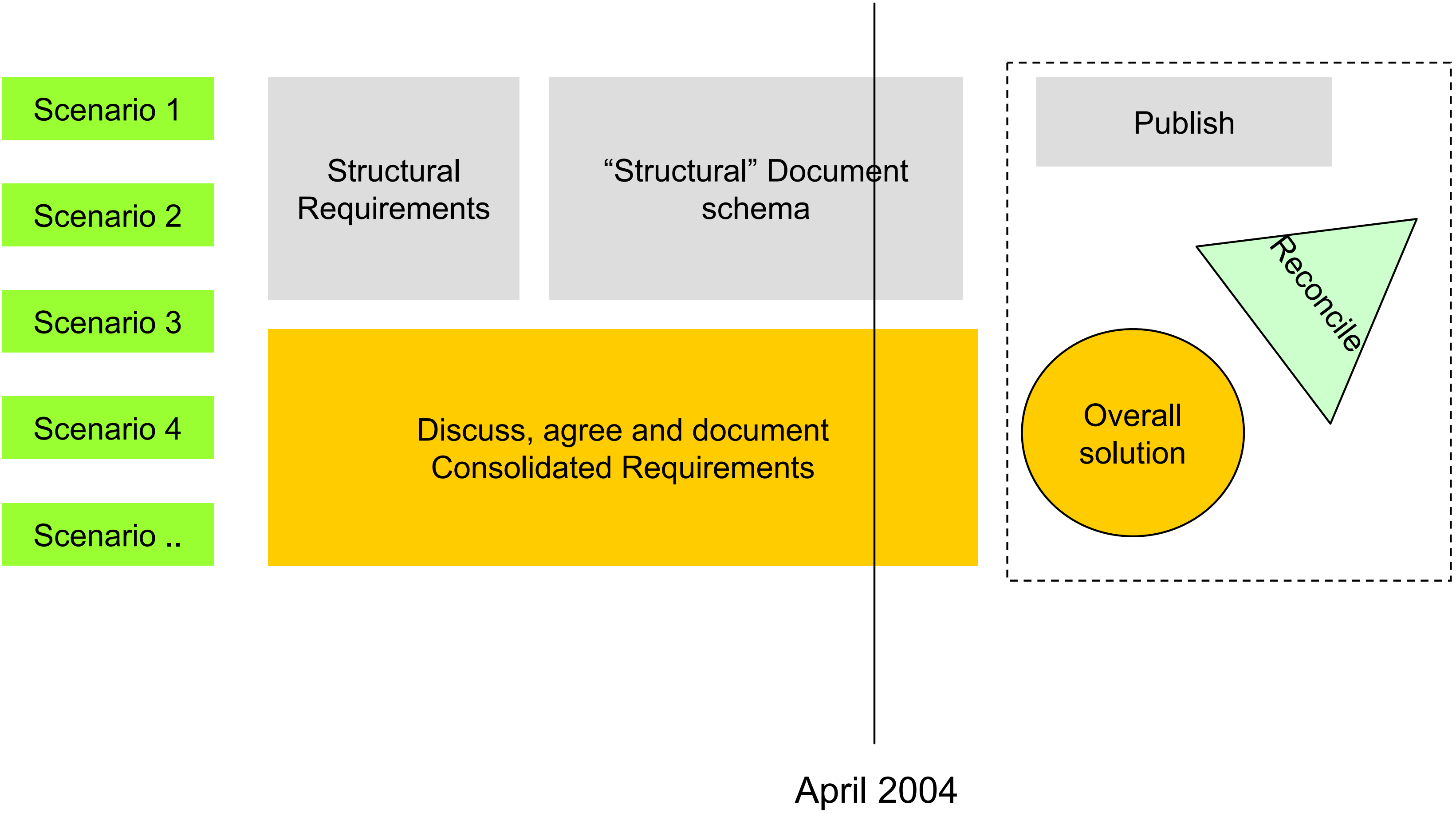
Americans today choose among more options in more parts of life than has ever been possible before. To an extent, the opportunity to choose enhances our lives. It is only logical to think that if some choice is good, more is better; people who care about having infinite options will benefit from them, and those who do not can always just ignore the 273 versions of cereal they have never tried. Yet recent research strongly suggests that, *psych*ologically, this assumption is wrong. Although some choice is undoubtedly better than none, more is not always better than less.

SPEEDLEGAL

W3C XHTML2 Working Group

- Status of XHTML 2
- Feedback
- Conformance

eContracts – overall requirements



Scenario 1

Scenario 2

Scenario 3

Scenario 4

Scenario ..

Structural Requirements

"Structural" Document schema

Discuss, agree and document Consolidated Requirements

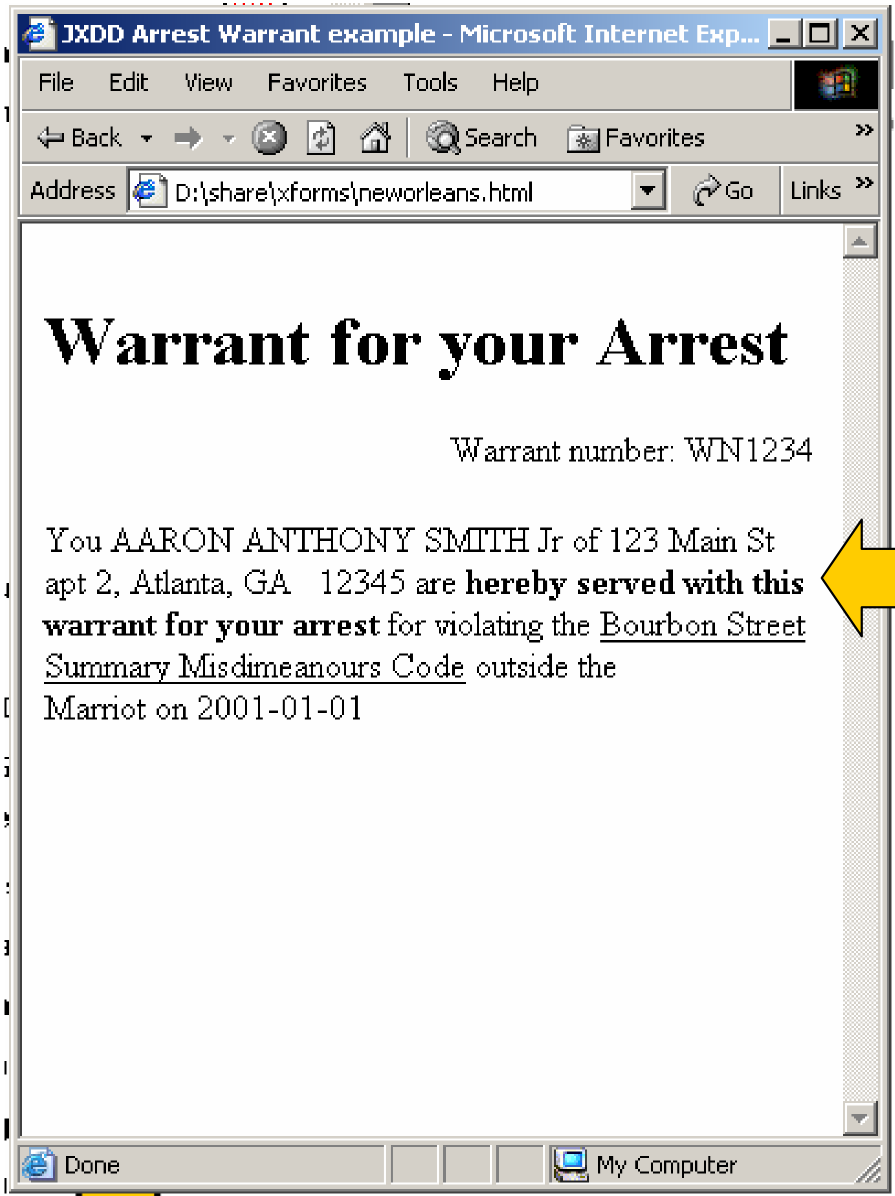
Publish

Reconcile

Overall solution

April 2004

Aside: XHTML2 technologies for semantics?


JXDD Arrest Warrant example - Microsoft Internet Exp...
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\share\xforms\neworleans.html Go Links
Done
My Computer

Warrant for your Arrest

Warrant number: WN1234

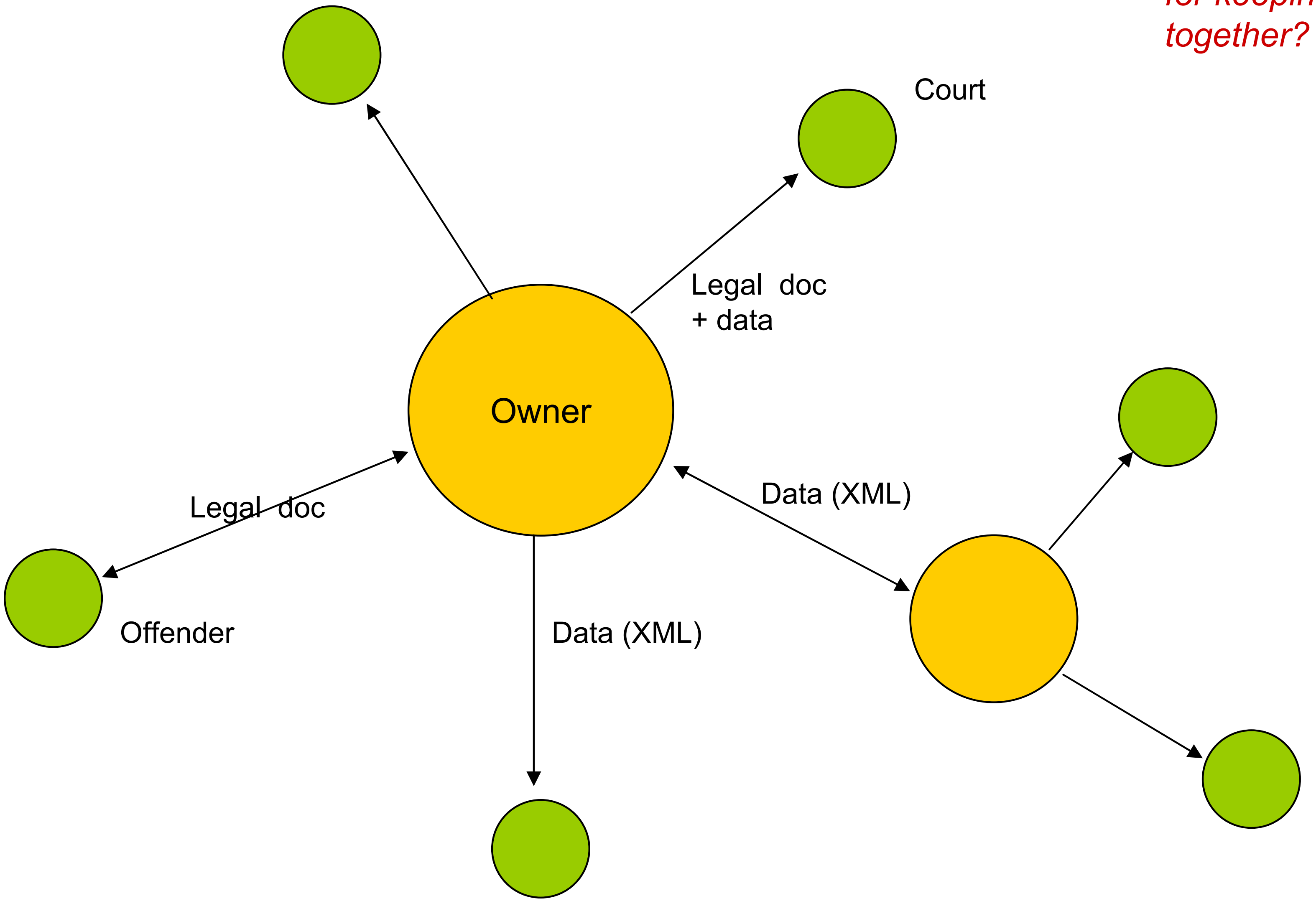
You AARON ANTHONY SMITH Jr of 123 Main St apt 2, Atlanta, GA 12345 are **hereby served with this warrant for your arrest** for violating the Bourbon Street Summary Misdimeanours Code outside the Marriot on 2001-01-01

??

```
<jxdd:ArrestWarrant xsi:type="aw:WarrantType">
  <!--jxdd:ActivityID>warrantNumber</jxdd:ActivityID-->
  <jxdd:ActivityID>WN1234</jxdd:ActivityID>
  <jxdd:ActivityCommentText>comment</jxdd:ActivityCommentText>
  <jxdd:CourtOrderDesignatedSubject xsi:type="aw:SubjectType" jxdd:id="Subject1">
    <jxdd:PersonName>
      <jxdd:PersonGivenName>AARON</jxdd:PersonGivenName>
      <jxdd:PersonMiddleName>ANTHONY</jxdd:PersonMiddleName>
      <jxdd:PersonSurName>SMITH</jxdd:PersonSurName>
      <jxdd:PersonSuffixName>Jr</jxdd:PersonSuffixName>
    </jxdd:PersonName>
    <jxdd:Residence>
      <jxdd:LocationAddress jxdd:id="Address1">
        <jxdd:AddressStreet>
          <jxdd:StreetFullText>123 Main St apt 2</jxdd:StreetFullText>
        </jxdd:AddressStreet>
        <jxdd:AddressCityName>Atlanta</jxdd:AddressCityName>
        <jxdd:AddressStateCode>
          <usps:USStateCodeTypeElement>GA</usps:USStateCodeTypeElement>
        </jxdd:AddressStateCode>
        <jxdd:AddressPostalCodeID>12345</jxdd:AddressPostalCodeID>
      </jxdd:LocationAddress>
      <jxdd:LocationContactInformation>
        <jxdd:ContactTelephoneNumber>
          <jxdd:TelephoneAreaCodeText />
          <jxdd:TelephoneExchangeText />
          <jxdd:TelephoneSubscriberText />
          <jxdd:TelephoneSuffixText />
        </jxdd:ContactTelephoneNumber>
      </jxdd:LocationContactInformation>
      <jxdd:ResidenceTypeText>apartment</jxdd:ResidenceTypeText>
      <jxdd:ResidenceDescriptionText>Offender is rooming with a friend.</jxdd:ResidenceDescriptionText>
      <jxdd:ResidenceStartDate>2001-10-10</jxdd:ResidenceStartDate>
    </jxdd:Residence>
    <jxdd:Employment>
      <jxdd:EmploymentEmployerName>employer</jxdd:EmploymentEmployerName>
      <jxdd:EmploymentOccupationText>occupation</jxdd:EmploymentOccupationText>
      <jxdd:EmploymentLocation />
    </jxdd:Employment>
    <jxdd:PersonBirthDate>1982-12-22</jxdd:PersonBirthDate>
    <jxdd:PersonBirthLocation />
    <jxdd:PersonAssignedIDDetails>
      <jxdd:PersonSSNID>270804583</jxdd:PersonSSNID>
      <jxdd:PersonDriverLicenseID jxdd:DrivingJurisdictionAuthorityCode="MN">MN 8989KJ7897</jxdd:Persor
      <jxdd:PersonFBIID>FBI 454545444</jxdd:PersonFBIID>
```

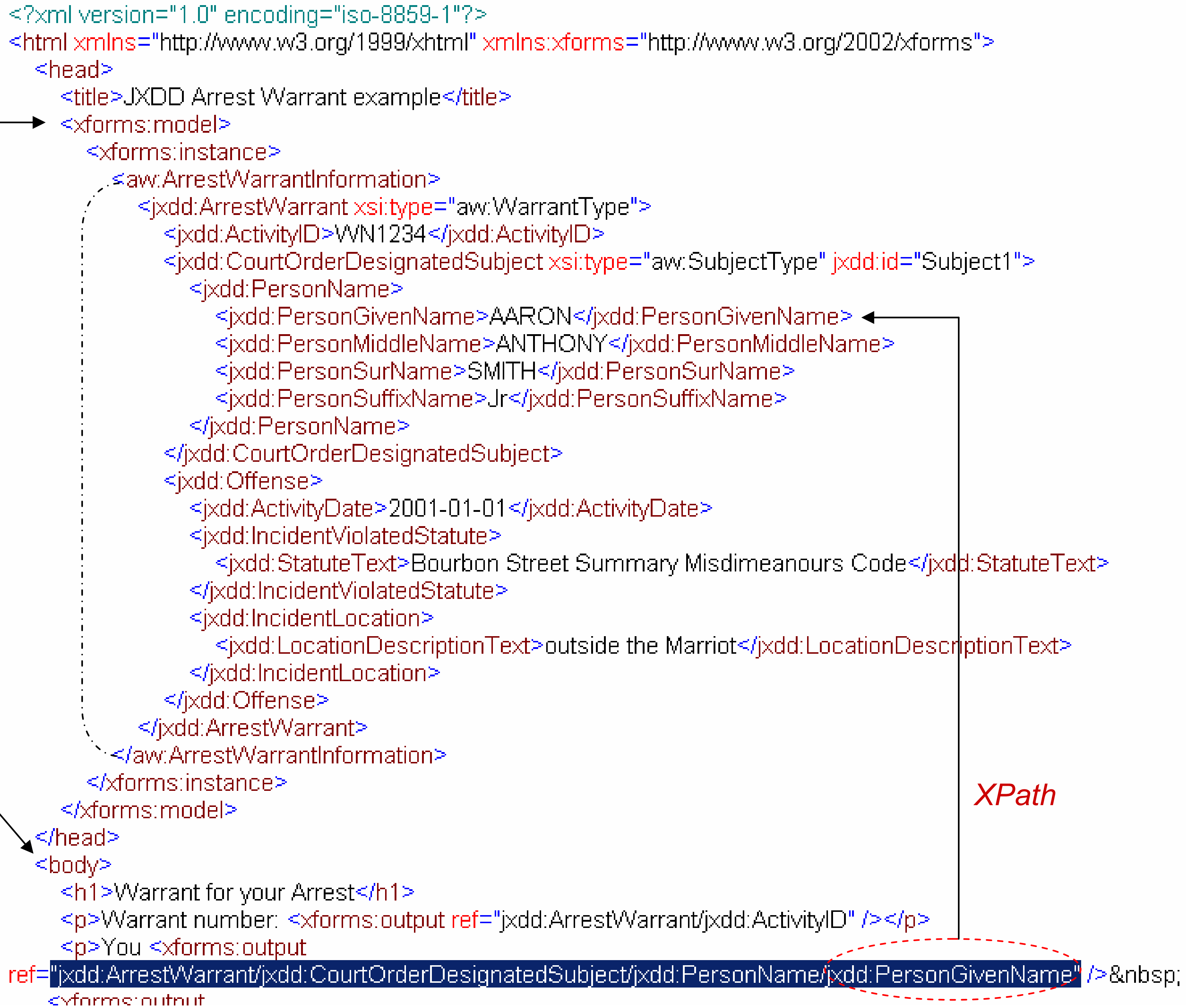
SPEEDLEGAL

Aside: XHTML2 technologies for semantics?



How compelling is the business case for keeping the two together?

Aside: XHTML2 technologies for semantics?



```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:xforms="http://www.w3.org/2002/xforms">
  <head>
    <title>JXDD Arrest Warrant example</title>
    <xforms:model>
      <xforms:instance>
        <aw:ArrestWarrantInformation>
          <jxdd:ArrestWarrant xsi:type="aw:WarrantType">
            <jxdd:ActivityID>WN1234</jxdd:ActivityID>
            <jxdd:CourtOrderDesignatedSubject xsi:type="aw:SubjectType" jxdd:id="Subject1">
              <jxdd:PersonName>
                <jxdd:PersonGivenName>AARON</jxdd:PersonGivenName>
                <jxdd:PersonMiddleName>ANTHONY</jxdd:PersonMiddleName>
                <jxdd:PersonSurName>SMITH</jxdd:PersonSurName>
                <jxdd:PersonSuffixName>Jr</jxdd:PersonSuffixName>
              </jxdd:PersonName>
            </jxdd:CourtOrderDesignatedSubject>
            <jxdd:Offense>
              <jxdd:ActivityDate>2001-01-01</jxdd:ActivityDate>
              <jxdd:IncidentViolatedStatute>
                <jxdd:StatuteText>Bourbon Street Summary Misdimeanours Code</jxdd:StatuteText>
              </jxdd:IncidentViolatedStatute>
              <jxdd:IncidentLocation>
                <jxdd:LocationDescriptionText>outside the Marriot</jxdd:LocationDescriptionText>
              </jxdd:IncidentLocation>
            </jxdd:Offense>
          </jxdd:ArrestWarrant>
        </aw:ArrestWarrantInformation>
      </xforms:instance>
    </xforms:model>
  </head>
  <body>
    <h1>Warrant for your Arrest</h1>
    <p>Warrant number: <xforms:output ref="jxdd:ArrestWarrant/jxdd:ActivityID" /></p>
    <p>You <xforms:output
ref="jxdd:ArrestWarrant/jxdd:CourtOrderDesignatedSubject/jxdd:PersonName/jxdd:PersonGivenName" />&nbsp;
  <xforms:output
```

"model"

document

XPath

Aside: XHTML2 technologies for semantics?

- RDF

Note: Mark Birbeck delivered presentation "RDF/XHTML: A New RDF Syntax" at XML Europe 2004, which presented a new (1 March) meta-data module for XHTML2 which he's been working on with Steven Pemberton that "makes it easy for processors to extract metadata as RDF triples, but without putting an unnecessary burden on authors familiar with HTML." See http://www.idealliance.org/papers/dx_xmle04/papers/04-04-02/04-04-02.html (slides not available as at 28 April)

Aside: XHTML2 technologies for semantics?

XForms

- “Standaside”
- No duplication
- Good for extracting data from a document, where you have a particular schema you want the data to fit
- Issues with fit for some data – is the schema a good match for the document? Is it acceptable to store data in the model which is invisible in the document?
- Enough tools exist to prove the concept
- .. But customised tools will be needed to create the standaside markup

RDF

- More “inline”
- Ease of authoring? To get attribute values:
  - either look up a dictionary of agreed terms (but must write authoring add-ins)
  - or constrain attribute values in schema (DTDs can’t do this adequately. Can W3C schema?)
- Maybe leave out of version 1, since tools don’t yet exist?
- FWIW, RDF can also be added using an XForms approach