

## John McClure

---

**To:** Ram Kumar  
**Cc:** David RR Webber - XMLGlobal; 'Mike Young'; 'Jeff Fisher'; 'Gabe Minton'; 'Todd Boyle'; Klaus-Dieter Naujok  
**Subject:** RE: Our semantics.

Hi Ram,

I'm not able to post to your committee's listserv, but here is my response just the same. And I've neglected to mention that I am speaking as the Architect for the [www.DataConsortium.org](http://www.DataConsortium.org), and the chair of two workgroups in LegalXML ([www.LegalXML.org/Contracts](http://www.LegalXML.org/Contracts) and [www.LegalXML.org/Dictionary](http://www.LegalXML.org/Dictionary)).

Since blind copies are being sent to our listservs, I've appended the original note I sent you.

Thanks,  
John McClure  
Hypergrove Engineering  
211 Taylor Street, Suite 32-A  
Port Townsend, WA 98368  
360-379-3838 (land)

For a discussion group about the Data Consortium Namespace, please  
<http://groups.yahoo.com/group/DCNArchitecture/join>

bcc: HORIZONTAL WG; DCN Architecture; Joe Reagle; Tim Berners-Lee; Murk Muller

> -----Original Message-----

> From: David RR Webber - XMLGlobal [mailto:Gnosis\_@compuserve.com]  
> Sent: Wednesday, October 03, 2001 9:15 AM  
> To: Ram Kumar  
> Cc: 'John McClure'; CIQ TC (E-mail); 'Mike Young'; 'Jeff Fisher'; 'Gabe Minton'; 'Todd Boyle'  
> Subject: RE: Our semantics.

>  
> Ram,  
>  
> It seems to me that the approach John is promoting is  
> counter to what people expect from a 'natural' use of  
> the expressive structure of XML. The current CIQ address  
> definitely follows that natural use model.

Compare for yourself the impact of unregulated structure. The following is XScript encoding, ultimately converted by a preprocessor into some XML dialect, such as the Data Consortium's simple DTD containing just 50 or so XML elements. This pre-processing is necessary in order to establish a stable environment for XPath-based processing, such as XSLT. Anyway, here's the XScript, containing just 11 elements. These elements are all named the same as if specified in a scripting environment (such as that provided by the DC's open-source developer toolkit).

```
<Addressee.Person>  
  <PersonalTitle.Abbreviation>Mr.</PersonalTitle.Abbreviation>  
  <GivenName>Ram</GivenName>  
  <OtherName.Initials>V</OtherName.Initials>  
  <FamilyName>Kumar</FamilyName>  
  <DeliveryAddress.SecondaryAddressee value='Privacy Link Proprietary Limited'/>  
  <DeliveryAddress.PostalBox.Title>PO Box 773</DeliveryAddress.PostalBox.Title>  
  <DeliveryAddress.PostOffice.Title>Chatswood</DeliveryAddress.PostOffice.Title>  
  <DeliveryAddress.PostalDistrict.Title.Abbreviation value='NSW'/>  
  <DeliveryAddress.PostalZone.Identifier>2057</DeliveryAddress.PostalZone.Identifier>  
  <DeliveryAddress.Country.Title>Australia</DeliveryAddress.Country.Title>  
</Addressee.Person>
```

The above XScript is equivalent to these 24 nested, difficult-to-quickly-grasp, XML elements. (This sample is normative material from the committee's spec.)

```
<Record>
  <xNL>
    <NameDetails NameType="Person">
      <PersonNameDetails>
        <Title>Mr</Title>
        <FirstNameDetails Type="GivenName">
          <FirstName>Ram</FirstName>
        </FirstNameDetails>
        <MiddleName Type="Initial">V</MiddleName>
        <LastName Type="SurName">Kumar</LastName>
      </PersonNameDetails>
      <DependencyNameDetails DependencyType="C/O">
        <NameDetails NameType="Organisation">
          <OrganisationName Type="Proprietary Limited">PrivacyLink</OrganisationName>
        </NameDetails>
      </DependencyNameDetails>
    </NameDetails>
  </xNL>
  <xAL>
    <!-- POBox: 773, Chatswood,NSW 2057, Australia -->
    <AddressDetails AddressType="Postal">
      <Country>
        <CountryName>Australia</CountryName>
        <AdministrativeArea Type="State">
          <AdministrativeAreaName>NSW</AdministrativeAreaName>
          <Locality>
            <LocalityName>CHATSWOOD</LocalityName>
            <PostBox Type="POBox">
              <PostBoxNumber>773</PostBoxNumber>
              <PostalCode>
                <PostalCodeNumber>2057</PostalCodeNumber>
              </PostalCode>
            </PostBox>
          </Locality>
        </AdministrativeArea>
      </Country>
    </AddressDetails>
  </xAL>
</Record>
```

>  
> Also - the ebXML approach is designed to move the  
> semantic clutter OUT of the transactional markup and  
> into core component definitions accessible via a  
> registry and cross referenced by UID.

The DC's RDF-based dictionary contains all this 'clutter' -- it is accessible not through Registry APIs but rather through the Data Consortium's scripting language, XScript. And I should also point out the semantic clutter (you and I dislike in transactional streams) does in fact exist in the CIQ markup but not in ours, i.e., all that 'typing' information that has been placed in attributes. As a sidebar, we uniquely identify objects using the (standard) rdf:ID attribute, not XML's id attribute, preferring to preserve the XML's id attribute for its conventional purpose -- resolution of intra-datastream references. Yes, the value for the rdf:ID attribute can be a URN, but for dictionary-based metadata, the W3C's XML Base standard is used. Thus it may be concluded that we are more interested in specifying a controlled vocabulary rather than a database. At the same time, it is certainly ok to use URNs as the value of the rdf:ID for instances, though the DC hasn't yet fully explored the impacts given our business context.

>

- > This nets huge definite benefits across the board
- > in simplicity, ease of use, maintenance and above all
- > abstracts the business semantics away from any
- > flavour of the month - whether it be RDF, Semantic Web
- > or whatever - AND gets you language independence.

I agree very much with the need to, "above all abstract the business semantics away from any flavor of the month". The difference here is that (as said in the covering note) "XScript is an ECMA-compliant front-end to XML-encoded datastreams, thus insulating Data Consortium members from changes in the encoding of those streams." So, while it seems your stakeholders want protection from changes in W3C standards, we adopt those standards (and also ECMA standards) in order to protect our stakeholders from changes in ebXML, UBL, cXML, and the others. In other words, we protect software against those standards judged to be comparatively more volatile, and we have judged W3C metadata and namespace standards to be less volatile over the next few years.

The DC Dictionary does provide language-independence via the XScript layer, using a single language (English) for the underlying native-XML representation processed by XPath. In fact, pre-processing is not at all an unusual thing for a vendor or corporate organization to do, so the question becomes how best interchange standards can leverage that natural system requirement. The DC rejects the notion of writing stylesheets geared to UIDs rather than natural language-based tag-names -- we are concerned about debugging such monstrosities on a planetary scale. We believe that it would be a mess for stylesheets to deal with tags that can be in multiple languages!

- >
- > An obvious next step for CIQ is to develop ebXML core
- > components. We will be able to do that very shortly once
- > the XBDL work standardizes on a representation model
- > that people can submit using. I expect this to happen
- > over the next month - a white paper will be out next
- > week.

Perhaps the 'core component' that I'm looking for, in order to make direct comparisons, is what we've simply called a "DeliveryAddress". In the DC's Dictionary, "DeliveryAddress" is a subtype of a Topic resource-type, thus we're positioned for adoption of Topic Map architectures. Context is handled through the RDF's subtyping mechanism, so that for instance, a HungarianDeliveryAddress subtype of DeliveryAddress could be established, specifying properties unique to it over the inherited class, and those properties in its inherited class whose values are 'replaced' by the HungarianDeliveryAddress type. We also allow a datastream publisher to create their own RDF dictionary, as one-offs of the DC's dictionary of course, thus handling organization-specific "context".

But I guess the most glaring difference is that by adopting an RDF orientation in our standards, we are able to assign multiple types to any object; I haven't seen any examples whatsoever how the following can be encoded as simply, as regularly, and as elegantly, as is done under the Resource Description Framework:

```
<Person>
  <rdf:type rdf:resource='Man'/>
  <rdf:type rdf:resource='DivorcedIndividual'/>
  <rdf:type rdf:resource='BrazilianCitizen'/>
  <rdf:type rdf:resource='DisabledPerson'/>
  <rdf:type rdf:resource='AverageWeight'/>
  ... other characterizations of the "Person" ...
</Person>
```

- > Therefore I do not see a need to change our current
- > approach.
- >
- > Thanks, DW.
- > =====
- > Message text written by Ram Kumar
- >
- > Thanks for the info. Appreciated. I will go through your doc.
- > and will get back to you on your suggestions.
- >
- > Regards
- > Ram

-----Original Message-----

From: John McClure [mailto:hypergrove@olympus.net]

Sent: Tuesday, October 02, 2001 12:17 PM

To: rkumar@msi.com.au

Cc: Todd Boyle; Gabe Minton; Jeff Fisher; Mike Young; vincent.buller@and.com

Subject: Addressing - Using the Data Consortium Namespace (DCN)

Mr. Ram Kumar, Chair

OASIS Customer Information Quality Committee

<http://www.oasis-open.org/committees/ciq>

Mr. Kumar,

Attached is a document containing examples of Data Consortium Namespace (DCN) encoding for addresses. Your technical committee is concerned with addresses, so I thought you might have feedback about our approach, since we are using a "dotted-tag". This document contains the DCN's encoding for samples that were posted on your website prior to your current specification (which has many more samples). The DCN's approach is one that appears less complicated than what the technical committee has now published, but I know already that some functional differences do exist when comparing the two. However, the DCN approach appears less complicated in part because of our use of a "dotted-tag" means that much less element nesting occurs. (In the Data Consortium, we have found that useability of a schema increases as nesting is reduced. Our encoding seems to be "about right" for the needs of Data Consortium members. You might have a wholly different opinion though.)

A "dotted-tag" is one that combines two adjacent tags, separating them by a 'period' -- it's handy because two adjacent nouns often are adequate to imply a connecting verb, and therefore DCN datastreams can conform to the Resource Description Framework (RDF) in an automated way. However, it also means that a pre-processor needs to convert a datastream into a fixed representation for querying by XSL stylesheets, because the way that dotted tags are encoded is entirely under the control of the publisher - adjacent tags are meant to be arbitrarily combined by the publisher. In the DC, we define these 'dotted-tags' in a specification for what we call "XScript". Basically, XScript is an ECMA-compliant front-end to XML-encoded datastreams, thus insulating Data Consortium members from changes in the encoding of those streams.

What's here is not the entire DCN picture, since our implementation of the standards for secondary address types as defined by the US Postal Service (e.g., basement apartments) is not readily apparent. To support secondary address types, we define appropriate object classes in our dictionary, and then weave those classes into the content models defined by our XScript specification.

Feel free to redistribute this information to your mailing list. I hope you find this helpful to your work, and we look forward to your comments and suggestions.

Thanks,

John McClure