11. Models of Business Information

DE + IA (INFO 243) - 27 February 2008
Bob Glushko

Plan for Today's Class

XML Vocabularies
- XMLification and EDI {and, or, vs} XML
- Vertical vs Horizontal Vocabularies

OASIS, Cover Pages, and other resources
EDI and XMLification

EDI took hold in the 1980s and 1990s and is widely used to automate routine document exchange transactions between established trading partners, especially in direct procurement and in the demand chains for cpg sector ("Consumer Packaged Goods")

After XML's emergence in the late 1990s, many standards efforts have arisen to evolve document and process standards encoded in EDI syntaxes to XML

EDI isn't dead, but XMLification is inevitable because EDI's implementation and recurring costs are vastly higher than those for XML-based information systems
X12 EDI Message Fragment

ISA*00* *00* *08*JCPenny111 *ZZ*test22222 *971107*1220*U*00302*112240001*1*P*@
GS*PO*test111111*test22222*19971107*1220*123456789*X*003042
ST*850*4567
BEGIN*00*NE*10117564**19990105
REF*Z1*11
REF*1V*97-0049393
PER*OD*Chris Smith*EM*csmith@supplyworks.com
PER*RE*Tom Gerry*TE*617-861-7900x1567
DTM*074*19990120*1806
N1*BE*SupplyWorks*91*FFF
N1*BY*SupplyWorks*91*ABC1235
N1*EO*Chris Smith*92*csmith1
N2*Room 208C*SupplyWorks Corporate
N3*57 Bedford Street*Building 2
N4*Lexington*MA*02173
N1*ST*SupplyWorks*ZZ*SW1
N1*SF*EC Office Supplies*1*98629321
N2*Purchase Department
N3*123 Main Street
N4*NY*NY*03417
PO1*1*24*EA*1.86**UI*999999922234
TXI*TX*9.41
PO1*2*12*CA*25.67**UI*999999955567
TXI*TX*4.51
CIT*2*42
SE*24*4567
GE*1*123456789
IA=1*112240001

EDIFACT Message Fragment

20000300:302*DTM+158:20000305:102*DTM+159:20000722:102*NAD+SU+9876543
N1*888836::92*GIS+37*NAD+ST+72661::92*LIN+++93258944:IN'PIA+1:8
04*REF*ON:XXXX00004*QTY+79:6660:EA'DTM+51:19991225:102'DTM+52:20000304
2000313:102'SCC4++W:16'QTY+1:900:EA'DTM+158:20000320:102'SQTY+1:900:
2000016:102'SCC3*QTY+3:17655:EA'DTM+51:19991225:102'DTM+52:20000525:
0:EA'UNT+73+770001'UNZ+1+77'UNS+UNOA+2:BFT;ZZ:CAI:ZZ+00305:2338+78++
**XML Message Fragment**

http://docs.oasis-open.org/ubl/os-UBL-2.0/xml/UBL-Order-2.0-Example.xml

```xml
<cac:OrderLine>
   <cbc:Note>this is an illustrative order line</cbc:Note>
   <cac:LineItem>
      <cbc:ID>1</cbc:ID>
      <cbc:SalesOrderID>A</cbc:SalesOrderID>
      <cbc:Quantity unitCode="KG">100</cbc:Quantity>
      <cbc:LineExtensionAmount currencyID="GBP">100.00</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="GBP">17.50</cbc:TotalTaxAmount>
      <cac:Price>
         <cbc:PriceAmount currencyID="GBP">100.00</cbc:PriceAmount>
         <cbc:BaseQuantity unitCode="KG">1</cbc:BaseQuantity>
      </cac:Price>
      <cbc:Description>Acme beeswax</cbc:Description>
      <cbc:Name>beeswax</cbc:Name>
   </cac:LineItem>
   <cac:Price>
      <cbc:PriceAmount currencyID="GBP">100.00</cbc:PriceAmount>
      <cbc:BaseQuantity unitCode="KG">1</cbc:BaseQuantity>
   </cac:Price>
</cac:OrderLine>
```

**Why You'd Encode A New Vocabulary in XML and Not EDI**

![Graph showing the benefit of using XML syntax over EDI over time with increasing implementation and maintenance cost.](image)
**Why EDI Hasn't Gone Away**

---

**Implementation & Maintenance Cost**

---

**EDI**

---

**XML**

---

*Cost of creating XML document types and mapping to/from EDI*

---

(XML) Vocabularies

---

The most accessible information pattern resources are XML vocabularies (or "schemas" or "tag sets")

A vocabulary is a set of elements and attributes and the rules by which they combine

Linguists shudder at this definition because it conflates the vocabulary and the grammar

XML purists call these "applications" of XML but this conflicts with the more common usage of that word to mean "software application"
The Need for Controlled Vocabularies

There is no one good access term or name for most objects or concepts. The idea of an "obvious, "self-evident," or "natural" term is a myth.

The words people use to describe things or concepts are "embodied" in their context and experiences... so they are often different or even "bad" with respect to the words used by others.

What would "good" words be like?

Where would they come from?

How do you get people to use them?

Why XML Vocabularies Exist

Vocabularies must exist because:

- There is no predefined set of tags
- And since there is no predefined tag set, there can't be any predefined semantics
- Semantics are defined in the schema and its documentation about what the elements and attributes mean
- So using XML without a (documented) schema makes little sense
Why Vocabularies Are Desirable

Vocabularies are desirable because people want to have a set of tags that is customized to their problem or industry so that they can take full advantage of XML.

A good vocabulary represents a significant investment in defining a domain, identifying its key semantic components, and specifying the constraints and rules governing the combination and reuse of those components.

A good vocabulary is a reference model for the domain that facilitates communication between enterprises operating within it.

The Best Thing About XML

Is the ease with which you can create a new vocabulary

The key word here is "you"
The Worst Thing About XML

Is the same as the best thing: the ease with which you can create a new vocabulary

There are often multiple vocabularies for the same or related domains and especially for the common information models that are used in more than one domain

That two concepts use the same XML tag names doesn't prove anything; the same content will inevitably be described using different names, and different content will be given the same names

Vertical and Horizontal Vocabularies

Vertical:
- Particular industry or vertical market
- Detailed product semantics
- Specialized process semantics
- Sometimes called "domain-specific" languages

Horizontal
- Concepts that are common to all (or a large number of) vocabularies
Creating Vertical XML Vocabularies – Worst Practices

Turn proprietary APIs into XML vocabularies by wrapping "<" and ">" around the names of methods that set and return values.

Turn proprietary database schemas into XML vocabularies by wrapping "<" and ">" around the names that define the structure of a record or object.

Turn existing EDI messages into XML by automated conversion, using the delimiters for segments, composites, and elements as "handles" for content.

Creating Vertical XML Vocabularies – Better Practices

Turn existing EDI messages into XML:
- Analyze EDI messages to identify the "syntax-neutral" conceptual models they contain.
- Encode these conceptual models in XML.

If no vocabulary exists:
- Identify current and potential uses of the vocabulary.
- Analyze existing documents and information sources (with EDI, identify "syntax-neutral" models of message content).
- Design conceptual models that satisfy the requirements in a feasible way.
- Encode the models in XML schemas.
**XML and Metcalfe's Law**

The value of a language depends on how many people (or computers) understand it.

How do you encourage and enable others to understand your language?

- Standardization Approach 1: "Understand my language or I won't do business with you"
- Standardization Approach 2: "Excuse me, here's my language, would you like to do business with me?"

---

**The Interoperability Problem**

The vocabulary problem implies an *interoperability* problem.

This means that two applications or services can't use each other's models or document instances "as is".

Some interoperability problems can be detected and resolved by completely automated mechanisms.

Other problems can be detected and resolved with some human intervention.

Other problems can be detected but not resolved.

Some problems can go undetected.
Syntactic and Semantic Interoperability

Syntactic interoperability is just the ability to exchange information. It requires agreement or compatibility at the transport and application layers of the communications protocol stack, with the messaging protocol and format, and with messaging choreography / sequencing.

Syntactic interoperability is necessary but not sufficient.

Semantic interoperability requires that the content of the message be understood by the recipient application or process.

The E-Business "Standards Pyramid"
Why Semantic Interoperability Problems Are Often Inevitable

Each new vocabulary for a particular industry is a step forward for that community, but proliferates definitions of information models that are common to many of them.

Since the distinctive or specialized parts of each vocabulary are the industry-specific "vertical" parts, a lot of attention gets paid to them.

In contrast, relatively less effort is given to the "horizontal" parts that seem more familiar or understandable.

Nevertheless, any large company – even highly verticalized ones – engages in diverse business activities that require it to understand multiple vocabularies at different times.

Vertical and Horizontal Vocabularies Must Work Together

[Diagram showing the relationship between automotive and chemical vocabularies, highlighting basic structures and industry-specific extensions]
When Models Don't Match

Suppose you publish your web service interface description and tell the world "my ordering service requires a purchase order that conforms to this schema"

This says "send me MY purchase order" not "send me YOUR purchase order"

How likely is it that the purchase orders being used by other firms will be able to meet your interface requirement, either directly or after being transformed?

How Bad Can the Interoperability Problem Be?

*Problem:* Does my "purchase order" mean the same thing as everyone else's?
The Target Model For The Interoperability Scenarios

The XSD Schema for the Expected Order [1]

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="Order" type="OrderType"/>
    <xs:complexType name="OrderType">
        <xs:sequence>
            <xs:element name="BuyersID" type="xs:string"/>
            <xs:element name="BuyerParty" type="PartyType"/>
            <xs:element name="OrderLine" type="OrderLineType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PartyType">
        <xs:sequence>
            <xs:element name="ID" type="xs:string"/>
            <xs:element name="PartyName" type="PartyNameType"/>
            <xs:element name="Address" type="AddressType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PartyNameType">
        <xs:sequence>
            <xs:element name="Name" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:schema xml:space="preserve"/>
</xs:schema>
```
The XSD Schema for the Expected Order [2]

```xml
<xs:complexType name="AddressType">
  <xs:sequence>
    <xs:element name="Room" type="xs:string"/>
    <xs:element name="BuildingNumber" type="xs:string"/>
    <xs:element name="StreetName" type="xs:string"/>
    <xs:element name="CityName" type="xs:string"/>
    <xs:element name="PostalZone" type="xs:string"/>
    <xs:element name="CountrySubentity" type="xs:string"/>
    <xs:element name="Country" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OrderLineType">
  <xs:sequence>
    <xs:element name="LineItem" type="LineItemType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="LineItemType">
  <xs:sequence>
    <xs:element name="BookItem" type="BookItemType"/>
    <xs:element name="BasePrice" type="xs:decimal"/>
    <xs:element name="Quantity" type="xs:int"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BookItemType">
  <xs:sequence>
    <xs:element name="Title" type="xs:string"/>
    <xs:element name="Author" type="xs:string"/>
    <xs:element name="ISBN" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

The Expected Instance

```xml
<Order>
  <BuyersID>91604</BuyersID>
  <BuyerParty>
    <ID>KEEN</ID>
    <PartyName>
      <Name>Maynard James Keenan</Name>
    </PartyName>
  </BuyerParty>
  <Address>
    <Room>505</Room>
    <BuildingNumber>11271</BuildingNumber>
    <StreetName>Ventura Blvd.</StreetName>
    <CityName>Studio City</CityName>
    <PostalZone>91604</PostalZone>
    <CountrySubentity>California</CountrySubentity>
    <Country>USA</Country>
  </Address>
  <OrderLine>
    <LineItem>
      <BookItem>
        <Title>Foucault's Pendulum</Title>
        <Author>Umberto Eco</Author>
      </BookItem>
      <BasePrice>7.99</BasePrice>
      <Quantity>1</Quantity>
    </LineItem>
  </OrderLine>
</Order>
```
Identical Model with Different Tag Names

[1]  

```xml
<Customer>
  <Number>KEEN</Number>
  <Name>
    <BusinessName>Maynard James Keenan</BusinessName>
  </Name>
  <Location>
    <Unit>505</Unit>
    <StreetNumber>11271</StreetNumber>
    <Street>Ventura Blvd.</Street>
    <City>Studio City</City>
    <ZipCode>91604</ZipCode>
    <State>California</State>
    <Country>USA</Country>
  </Location>
</Customer>
```

Identical Model with Different Tag Names

[2]  

```xml
<Acheteur>
  <ID>KEEN</ID>
  <Nom>
    <NomCommercial>Maynard James Keenan</NomCommercial>
  </Nom>
  <Adresse>
    <Appartment>505</Appartment>
    <Bâtiment>11271</Bâtiment>
    <Rue>Ventura Blvd.</Rue>
    <Ville>Studio City</Ville>
    <CodePostal>91604</CodePostal>
    <Etat>California</Etat>
    <Pays>USA</Pays>
  </Adresse>
</Acheteur>
```
Same Model, Attributes Instead of Elements

```xml
<BuyerParty
  ID="KEEN"
  Name="Maynard James Keenan"
  Room="505"
  BuildingNumber="11271"
  StreetName="Ventura Blvd."
  City="Studio City"
  State="California"
  PostalCode="91604"
/>
```

Granularity Conflicts

```xml
<Address>
  <StreetAddress>11271 Ventura Blvd. #505</StreetAddress>
  <City>Studio City 91604</City>
  <CountrySubentity>California</CountrySubentity>
  <Country>USA</Country>
</Address>

<PartyName>
  <FamilyName>Keenan</FamilyName>
  <MiddleName>James</MiddleName>
  <FirstName>Maynard</FirstName>
</PartyName>
```
Assembly Mismatch - Separate Customer and Order Documents [1]

```xml
<BuyerParty>
  <ID>KEEN</ID>
  <PartyName>
    <Name>Maynard James Keenan</Name>
  </PartyName>
  <Address>
    <Room>505</Room>
    <BuildingNumber>11271</BuildingNumber>
    <StreetName>Ventura Blvd.</StreetName>
    <CityName>Studio City</CityName>
    <PostalZone>91604</PostalZone>
    <CountrySubentity>California</CountrySubentity>
    <Country>USA</Country>
  </Address>
</BuyerParty>
```

Assembly Mismatch - Separate Customer and Order Documents [2]

```xml
<Order>
  <BuyersID>91604</BuyersID>
  <BuyerParty>
    <ID>KEEN</ID>
  </BuyerParty>
  <OrderLine>
    <LineItem>
      <BookItem>
        <Title>Foucault's Pendulum</Title>
        <Author>Umberto Eco</Author>
      </BookItem>
      <BasePrice>7.99</BasePrice>
      <Quantity>1</Quantity>
    </LineItem>
  </OrderLine>
</Order>
```
Conceptual Incompatibility

<Address>
  <Latitude direction="N">37.871</Latitude>
  <Longitude direction="W">-122.271</Longitude>
</Address>

Lessons from the Interoperability Examples

There are a large number of ways that two implementation models that are supposed to be equivalent can fail that test.

But no matter how different they look, with different syntaxes, tag names, or assembly models, if their conceptual model is the same, it is possible to transform one implementation model to another.

Validation is not sufficient to guarantee complete interoperability.
Information Pattern Resources for Scavenger Hunt

Cover Pages

OASIS

- Compare to W3C and WS-I
- The Universal Business Language (UBL) is an OASIS TC

For Monday March 3

"Operation Clean Data" Malcolm Wheatley, CIO (July 2004)


"Key Issues of Technical Interoperability Solutions in eHealth and the RIDE Project" Asuman Dogac, Tuncay Namli, Alper Okcan, Gokce Laleci, Yildiray Kabak and Marco Eichelberg.