11. Information Integration and Interoperability

INFO 202 - 6 October 2008

Bob Glushko

Plan for INFO Lecture #11

Understanding the "Integration Problem"
"When Models Don't Match"
Enterprise Data Integration
Business Intelligence and "Dashboards"
Enterprise Information & Data Management Goals

Run the business more efficiently through greater automation and "straight through" or "end-to-end" processing of the information that it creates and receives

Consolidation of data from multiple business units to create a unified view of the {customer, supply chain, etc}

Get end-to-end visibility of business processes

Take different perspectives (from high level aggregation to resolving individual data anomalies or inconsistency)

Make better decisions more quickly

Enterprise (and Inter-enterprise) Information & Data Management Challenges

Internal to a firm, application "silos" or "stovepipes" may have been created over time and not have been designed to share information with each other

Each of these systems has a specific purpose and a data model customized for that purpose - so these models may be incomplete or incompatible with respect to each other

This causes problems when business processes can span multiple departments, business applications, or even multiple firms

These problems are greater when information or data comes from outside the firm
The Integration Requirement

Companies have so many internal (with employees) and external (with customers and suppliers) interactions that they must automate as many as possible.

This requires INTEGRATION -- the controlled and automated sharing of content, data and business processes among any services, applications, or information sources, intra- or inter-company.

Integration has long been a substantial portion of the IT activities in many companies.

Integration Is NOT Interoperation

Integration means that one application can extract or obtain information from another one.

It doesn't mean that the information will work "as is" for the receiving application or service.

Different systems may use different formats for nominally the same data items (September 11, 2001, 9/11/2001 and 9-11-01; 11/09/01 in Europe).

Furthermore, there may be significant semantic differences between data items with the same name.
Syntactic, Structural, & 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 and with the messaging protocol and encoding format.

Structural interoperability means that all of the expected information components are present with the same arrangement and granularity.

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

Interoperability isn't All or None

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.
Transformation in Semantic Integration

Semantic integration is the process by which a common semantic "data model" or "object model" is created through transformation

- Easy transformations include field length, data type or unit conversion
- Product attributes can be extracted from text descriptions
- Classification into product categories (e.g., UN SPSC) can be performed using some form of "nearest neighbor" or principle components analysis

What's the most powerful semantic integration processor?
Who does the transformation - the sender or the receiver?

Catalog Integration

We've talked about the need to combine information sources with different data/document models and semantics in many previous lectures

Combining "catalog information" is a challenge e-businesses (Stonebraker & Hellerstein) but also for digital libraries

The goal for both is to create catalogs that describe actual and "virtual" resources that they can provide access to without needing to collect or control them directly
Technical Challenges Shared by Multi-Enterprise Content / Digital Libraries

- Information comes from owners with varying relationships with the integrator
- Information comes in different formats and semantics
- Information requires multiple schemas and multiple taxonomies

Technical Challenges that Contrast Multi-Enterprise Content / Digital Libraries

- Digital libraries are integrating metadata, while e-business applications are integrating both metadata and content from hybrid document types
- Standards for library content and metadata are better established than those for enterprise content
- Information is syndicated / personalized much more for enterprise content
- Enterprise information is operational and volatile
Non-Technical Challenges

Who makes decisions about destination formats?
Whose responsibility is it to create the destination format?
Who owns the new information?

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?
To Interoperate, or not to Interoperate?

**Problem:** Does my "purchase order" mean the same thing as everyone else's?

The Target Model For The Interoperability Scenarios

```
ORDER
   /
  /    *
PARTY ORDER LINE
   /
  /  LINE ITEM
 ADDRESS BOOK ITEM
   /
  /
 NAME
```
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: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>
</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>
    <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>
  <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>
    <PostalZone>91604</PostalZone>
    <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">
</BuyerParty>
```
Granularity Conflicts

Assembly Mismatch - Separate Customer and Order Documents [1]
Assembly Mismatch - Separate Customer and Order Documents [2]

<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.

The Dimensions of Interoperability

Agreement on Conceptual Model

Agreement on Implementation Model
The Dimensions of Interoperability

Agreement on Conceptual Model

<table>
<thead>
<tr>
<th>High</th>
<th>You can get there</th>
<th>You’re there</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>You can’t get there</td>
<td>Don’t be fooled</td>
</tr>
</tbody>
</table>

Agreement on Implementation Model

Low | High

An Enterprise Information Integration Scenario

An existing customer calls a service representative to increase an order

The service representative must:

- locate information about the customer
- locate the existing order
- determine if the order can be changed or whether a new order must be created
- determine whether to accept the order based on the customer's payment history and credit

What information sources or applications must the service rep consult? How can it be done?
Integration "By Eye"

Swivel-Chair Integration

The need to consult multiple unintegrated applications to locate information to complete a business process

Recent study by Corizon:

- 66% of call center agents use three applications or more to serve customers on a typical call
- 27% use five or more
- 71% claim time is wasted on or after a call because of switching between different applications
- 53% admit that errors creep in when entering data into multiple systems
Enterprise Portals

Adventure Portal Applications

Portal applications replace the different interfaces to multiple systems with a single, user-friendly screen that accesses only the parts of a back-end system that the employee needs.

Purpose is to create a unified experience with a "single sign-on"

You can think of this trying to recreate something like Yahoo for the enterprise (Intranet).

Nearly every major software vendor has created an enterprise portal solution that is an attempt to "up-sell" from the application server platform.
Enterprise Information Integration

Integration "by eye" is inadequate in situations with high transaction rates or complex data, and it is necessary for the applications to share data without human intervention.

This requires true semantic unification of the underlying logic and content models, which may or may not be presented to the user as a single "composite application".

For large firms, and increasingly for medium-sized ones, a solution is to implement an ERP system that integrates all of the operational data.

Business Intelligence

ERP and other enterprise systems contain the very granular and "live" operational data of the enterprise.

ERP systems generate historical reports that are useful for long-term decision making, but don't enable ad hoc analysis of operations needed to make tactical decisions.

So you need another set of your enterprise data organized in a data model optimized for asking questions rather than running your business.
Generic Enterprise Information Integration Architecture (Gantz, 2004)

Data Warehouses

A data warehouse is a "subject-oriented, integrated, time-varying, non-volatile collection of data used in organizational decision making"

Data warehouses extract data from ERP systems and other related business software applications into a separate repository

It is common practice to "stage" data prior to merging it into a data warehouse with an "Extract, Transform, and Load" (ETL) application

Since the information won't change, denormalization to improve query performance is a common ETL process

The data model for the warehouse, designed to enable efficient ad hoc data analysis and reporting, is sometimes called a "hypercube"

A common term for the analysis done in a warehouse is online analytical processing or OLAP
The Virtual Warehouse

A virtual warehouse is created "on demand" by centralizing and normalizing metadata about the data sources rather than the data itself.

The data is left in its original location and extracted only when needed, which makes more "real time" analysis and "business intelligence"
Driving Your Business

The best data warehouse design and the most clever OLAP won't help the business if the analysis can't be understood by the decision makers.

"Dashboards" combine information integration with information visualization to enhance the usability of business intelligence.

A dashboard provides hierarchical views appropriate to different management levels and the means to "drill down" to find details.

See idashboards.com or demo.visualmining.com

Dashboard UI for Business Intelligence
Readings for INFO Lecture #12

M. Brun, J. Brown and R. Lohde, "Adoption of UBL in Denmark: Business cases and experiences"

Smita Brunnermeier and Sheila Martin, "Interoperability Costs in the US Automotive Supply Chain"

Arnon Rosenthal, Len Seligman, and Scott Renner. "From Semantic Integration to Semantics Management: Case Studies and a Way Forward"