Semantic Data Modeling Within and Across The Firewall

Mark Crawford
Standards Architect
SAP Labs
mark.crawford@sap.com
The Data Interoperability Dilemma

**Users**

- Integrate with multiple processes and industries
- Do not want to learn, deploy and maintain different integration approaches

**Industry Communities**

- Want complete, interoperable specifications
- Want to avoid endless reinvention

**Internal Applications**

- Support data across multiple business functional areas
- Support interfaces with different databases and systems

**Application Providers**

- Support every industry and process
- Want easier, cheaper interoperability approaches to reduce TCO
How Serious is the Problem?

**Internal**
Data Modeling relies on the experiences and vocabularies of the modeler
Common Understanding and Naming is syntax and context specific
Little agreement across business functions and systems

**External**
More than 2000 different Business Standards dialects!
*Gartner:*
Only 5% of the interface is a function of the middleware choice
95% is a function of application semantics and logic
The Data Interoperability Dilemma

ISO 15000-5 Core Components

CCTS and The Semantic Web

Process Models, CCTS, and Syntax

CCTS in SAP

The Best Run Businesses Run SAP
Semantic Data Modeling Solution –
ISO 15000-5 Core Components

■ What:
  ■ A methodology for developing semantic data models

  ■ The integration mechanism for coordination between
    Architectures, Process Models, Data Models, and Syntax
    Expressions

  ■ A way to identify, capture and maximize the re-use of
    business information to support and enhance information
    interoperability across multiple business situations

■ How:

  ■ Syntax and Context Neutral conceptual models

  ■ Context specific physical/logical models

  ■ Transformation to syntax specific information exchanges
ISO 11179 –

- Describes the standardizing and registering of data elements to make data understandable and shareable
- Provides concrete guidance on the formulation and maintenance of discrete data element descriptions and semantic content – Identifiers, Definitions, & Classification Categories
- Used to formulate data elements in a consistent, standard manner
- Provides guidance for establishing a data element registry
ISO 15000-5 Benefits

- Enables interoperability among different industries and applications
- Holds related information together and avoids semantic fragmentation
- Facilitates multilingual support
- Syntax neutral
  - Models can be readily expressed in XML Schema, UML diagrams, JAVA classes, SQL based relational data bases, etc.
- Guarantees semantic understanding in any technical implementation and interface
- Does not require complicated and expensive mappings between interfaces.
- Significantly reduces data total cost of ownership
ISO 11179/ISO 15000-5 Naming Rules

**Semantic rules** – enable meaning to be conveyed
- Example: Components consist of discrete terms:
  - Object class terms, property terms, representation terms, qualifier terms

**Syntactic rules** – relate items in a consistent, specified order
- Example: a rule might require the property term is always the second component in the name

**Lexical rules** – (word form and vocabulary) rules reduce redundancy and increase precision
- Example: Nouns are used in the singular only

**Uniqueness rule** – ensures names are unique within a context.
- Example: Homonyms shall not occur in a namespace
  - bank (embankment)
  - bank (place where money is kept)
<table>
<thead>
<tr>
<th>ISO 11179 and ISO 15000-5 Named Constructs</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>11179</th>
<th>15000-5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object class</strong></td>
<td><strong>Address</strong></td>
</tr>
<tr>
<td>Property 1: representation 1</td>
<td>Street: text</td>
</tr>
<tr>
<td>Property 2: representation 2</td>
<td>Post code: text</td>
</tr>
<tr>
<td>Property 3: representation 3</td>
<td>Town: text</td>
</tr>
<tr>
<td>Property 4: representation 4</td>
<td>Country: identifier</td>
</tr>
</tbody>
</table>
The Core Component Model

- Core Components are the centerpiece of the approach
- Conceptual in nature
- Used as reference only
- Never instantiated
- Intended to have single global reference model
Aggregate Core Component (ACC)

A collection of related pieces of business information that together convey a distinct business meaning, independent of any specific Business Context.

Expressed in modelling terms, it is the representation of an Object Class, independent of any specific Business Context.

Core Components

- Aggregate Core Component
- Basic Core Components
  - Person. Details
  - Person. Given. Name
  - Person. Gender. Indicator
  - Person. Title. Text
- Association Core Component
  - Person. Address

One property must be present, either as a BCC or an ASCC.
Basic Core Component (BCC)

- Simple properties
- Equivalent to attribute of a class
- Key feature is consistent Tri-partite Naming
  - Object:Property:Representation

Class Diagram

- Class
- Attributes
  - Item
    - Description
    - Quantity
    - Identification

Core Components

- Aggregate Core Component
- Basic Core Components
  - Item. Details
  - Item. Description. Text
  - Item. Quantity. Quantity
  - Item. Identification. Identifier
Core Data Type

A Core Data Type signifies the “type” of information represented by a Basic Core Component.

Core Data Types do not contain business semantics – they are “semantic-neutral”

Dictionary Entry Name
Unqualified Data Types (CCTs + Secondary Representation Terms):
<Data Type Term>. Type

Qualified Data Types:
[?<Data Term Qualifier> _ *] <Data Type Term>. Type

22 Defined Data types

Unlimited Related Data types

Content Component carries the actual value

for creation of

Supplementary Component gives extra definition to the value
Complex Property

Naming mechanism for expressing the relationship between two object classes

Expresses the nature of the association

Class Diagram

Organization
  Name
  Identification

Aggregate Core Component

Organization
  Name
  Identification

Aggregate Core Component

Contact

Address
  Line One
  Line Two
  City Name
  Postal Code

Organization. Contact. Address
From Conceptual to Real World
The Advantage of Conceptual and Context Specific Semantics

Unambiguous understanding of business information in any industry (semantic) and application (technique)
Same use of business information internally (applications) and externally (between business partners)
Less effort for internal/external interoperability
Context Specific Artifacts - Business Information Entities

- BIEs are built from their corresponding CCs through the application of qualifiers and context.
- BIE composites are suitable for reuse with the same context values – as well as different context values.
- BIEs aggregate up to and including the message assembly level.
- BIEs specify restrictive data typing as appropriate for context specific instantiations.
- BIE models can be global, federated, or stand alone.
Differentiating Context Through Qualifiers

Qualifiers are used when the three data element name parts described earlier are not sufficient to uniquely identify a data element.

Qualifiers can be added in front of an Object Class, a Property Term, or both.

Examples:
- Office Address. City. Name
  - “Office” is a Qualifier for the “Address” Object Class
  - Differentiates this data element from one named Home Address. City. Name
- Organization. Primary Contact. Name
  - “Primary” is a Qualifier for the “Contact” Property Term
  - Differentiates this data element from one named Organization. Secondary Contact. Name

![Basic Business Information Entity Diagram]

Credit
Object Qualifier
- Object Property
- identifies the primary concept of a data element

Payment Card
Object Class
- identifies the primary concept of a data element

New
Property Qual.
- qualifies add. the characteristics of the object class

Expiration
Property Term
- identifies the characteristics of the object class

Date
Rep. Term
- categorizes the format of the data element

Object

Property
Differentiating Context Through 8 Space Values

Classification Scheme
- Context Category: String
- Definition: String
- Hierarchy: String
- Name: String
- Owner: String
- Primitive Type: String

Context Value
- Meaning: String
- Value: String

Business Context
- Business Process Context Value
- Business Process Role Context Value
- Geopolitical Context Value
- Industry Classification Context Value
- Official Contraints Context Value
- Product Classification Context Value
- Supporting Role Context Value
- System Capabilities Context Value

Common Information Registry Class
- Unique Identifier: String
- Version Identifier: String
Supporting Reuse: Metadata For Each Construct

- 9 Metadata Classes
- 35 Unique pieces of information
- Maps to ebXML RIM

Registery Class
- Unique Identifier: String
- Version Identifier: String

Status Information
- Comment: String [0..*]
- Reason: String [0..1]
- Reference: String [0..*]
- Start Date: Date
- Status: String

Administrative Information
- Registrar: String
- Registration Authority: String
- Submitting Organisation: String

Representation Information
- Constraint: String [0..*]
- Representation: String
- Representation Syntax: String

Replacement Information
- Replacement Date: Date
- Replacement Description: String

Change History
- Change Date: Date
- Change Description: String
- Change Type: Change Type
- Comment: String [0..*]
- Reference: String [0..*]
- Request By: String
- Request Date: Date

Descriptive Information
- Acronym: String [0..*]
- Comments: String [0..*]
- Keyword: String [0..*]
- Reference Document: String [0..*]

Association Information
- Association Multiplicity: Cardinality
- Association Type: Association Type
- Comment: String [0..*]
- Usage Description: String

Common Information
- Business Term: String [0..*]
- Definition: String
- Dictionary Entry Name: String
- Possible Value: String [0..*]
- Usage Rule: String [0..*]
The Data Interoperability Dilemma
ISO 15000-5 Core Components
Process Models, CCTS, and Syntax
CCTS and The Semantic Web
CCTS in SAP
Coordinated Standards Are Key

Stages of development over time

- Enterprise Architecture & Business Process Analysis
- Information Analysis/Data Modeling
- XML Development

Overlaps in standards

- Component Reuse
- Requirements
- Process Modeling
- Data Analysis and Design
- Syntax Expression

UML / UMM

ISO 11179 / ISO 15000-5

XML / XSD / NDR
UN/CEFACT Modelling Methodology (UMM)

Registry/Repository

UMM-Metamodel
- Business Domain View (BDV)
- Business Requirement View (BRV)
- Business Transaction View (BTV)
- Business Service View (BSV)

UMM-Workflows
- Business Modeling Workflow
- Requirements Workflow
- Analysis Workflow
- Design Workflow

Business Experts
Analysts
Designers

Worksheets
- Describe Business Domain Model
- Describe Business Area(s)
- Describe Process Area(s)
- Identify Business Process(es)
- REA-Worksheet
- Business Process
- Business Collaboration Specification
- Business Process Metric
- Business Collaboration
- B-Collaboration Lifecycle
- Business Entity
- Business Entity Lifecycle

- Business Collaboration Protocol
- Business Transaction
- Business Information

- Service Collaboration
- Business Objects

UML-Diagramtypes
- Area / Process Area (Package)
- Library Supported Business Processes (UseCase)
- Final BDV-Use Case Diagram (UseCase)
- Business Processes and Partners (UseCase)
- Business Process Activity Model (Activity)
- Conceptual Business Information Model (Class)
- Business Process Use Case
- Business Collaboration Use Case

- Business Collaboration Object Flow Diagram
- Use Case Diagram
- Business Transaction Object Flow Diagram
- Final Business Information Model

- Service Collaboration SequenceDiagram
- BusinessObject Class Diagram
Creating the Syntax – Optimized XSD

- Context Neutral Core Components provide the conceptual model
- Context specific BIEs provide the physical model
- XSD expressions following optimized Naming and Design Rules provide the syntax specific instantiations
- Tight integration between all three layers maximizes interoperability within and across industries
```
<xsd:element name="AvailabilityInventoryItem" type="cac:InventoryItemType"/>
<xsd:complexType name="InventoryItemType">
  <xsd:annotation>
    <xsd:documentation>
      <ccts:CategoryCode>ABIE</ccts:CategoryCode>
      <ccts:DictionaryEntryName>Inventory_Item. Details</ccts:DictionaryEntryName>
      <ccts:Definition>Details of an inventory item.</ccts:Definition>
      <ccts:ObjectClass>Item</ccts:ObjectClass>
      <ccts:RepresentationTerm/>
    </xsd:documentation>
  </xsd:annotation>
...
```

Synergy Between CCTS, OWL, RDF and Semantic Web

- RDF is Semantic Web locator
- OWL is Semantic Web language
- XML is expression of information
- CCTS can be heavy lifter at database level for easing burdens on XML/RDF/OWL
- Consistency to data structures, meaning and use
- Consistency in metadata

Semantic Web Stack

- Trust
- Proof
- Logic framework
- Signature
- Encryption
- Semantic Based Data
- Rule of OWL/Rule
- Schema
- XML
- Namespaces
- URI
- Unicode

Courtesy W3C
CCTS & RDF Alignment

- RDF Triple
  - Subject
  - Property
  - Value

- Core Component Triple
  - Object
  - Class
  - Property Term
  - Representation Term (Value Domain)
OWL Triples
- Classes (general things) in the many domains of interest
- The relationships that can exist among things
- The properties (or attributes) those things may have

CC Triples
- Object Class
- Associations
- Properties (Property Term + Representation Term)
<table>
<thead>
<tr>
<th>The Data Interoperability Dilemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 15000-5 Core Components</td>
</tr>
<tr>
<td>Process Models, CCTS, and Syntax</td>
</tr>
<tr>
<td>CCTS and The Semantic Web</td>
</tr>
<tr>
<td><strong>CCTS in SAP</strong></td>
</tr>
</tbody>
</table>
Enterprise Services (Platform View)

**Enterprise Standards**
- Reliability, scalability, performance, security
- Driven by business and legal requirements
- Mostly user-specific
- Examples: throughput of 100,000 invoices per hour, Section 508 accessibility

**Web Services Standards**
- SOA using Internet standards (XML, HTTP, etc.)
- Driven by interoperability needs with partner platforms (e.g. IBM WebSphere, Microsoft .NET)
- Covered by W3C, OASIS, WS-I
- Examples: WSDL, SOAP, WS-Security, WS-ReliableMessaging
Introducing Standards for Standards

Development Methodology Standards
- Modeling and design rules
- Driven by development and integration efficiency and quality needs
- Covered by OMG, UN/CEFACT
- Examples: UML, XML, CCTS, XML NDR

Effectively enhance reuse and address the Business Standards Dilemma

SAP is leveraging ISO 15000-5 CCTS and XML NDR for the semantic-based definition of Enterprise Services
- CCTS For Global Data Types
- XML NDR for XML internal and External expressions
Questions?
Take The Business Perspective

SOA IS REQUIRED BUT NOT ENOUGH

Common Business Semantics are required
**Platform Ecosystem**

**Collective Innovation**
- Customers, Partners, ISVs & SAP
- Develop and prioritize Enterprise Service Definitions

**Based on Open Standards**
- Interoperability: Web Services
- Methodology: CCTS, etc.