Initial Thoughts:

- Examine the differences between standards (vocabularies) and the EbXML Infrastructure
- Examine the differences between Core Components and Business Processes.
- Look at the overall architecture of ebXML
Goals:

- To provide a high level overview of how ebXML works
- Distinguish the run time viewpoints
- To show how different ebXML compliant Vocabularies can interoperate within the ebXML infrastructure.
Why XML

• Extensible Markup Language
• Not a fixed set of Elements (HTML)
• Allows data to be smart (declarative)
• Extensible (elements, namespaces)
• Widespread adoption & endorsement
• Interoperability is now possible
Business Process Models

Message

Business Content

Technical Content

Services Interface

Sends & Receives

Network Economy

Process

Services

Schemata

Codes, Elements

Feeds

Uses

Enables

Defines

Supports

Provides Access To

Uses

Populates

Defines

Metadata model to XML Transformation Rules

BP+R²

TRP

Common Technical Architecture

Distributed Repository

Registry

Security

Transport
TA Philosophy

Architect ebXML in Layers

The first layer has to recognize the Repository Items that constitute core components (ie address, tel)

The second Layer has to recognize the Business Process

Third Layer is discovery of what partners require

Current work focuses largely on Layer one!
At the heart of ebXML is a powerful system of Registries and Distributed Repositories.

Some repositories contain Core Components and some contain Syntax describing Business Process.

It is important that we can reference Items (CC) from Business Process Layer at the Element Level.
ebXML Architecture

- XML Elements in document instances contain pointers to RI's
- Repository Items are metadata – not instances of data

```xml
<?xml version="1.0">
<name ID="FooRep:12345">
  Duane
</name>
```

```xml
<?xml version="1.0">
<ID>12345</ID>
<Element>appelle</Element>
<EQ org="your_org">
  name
</EQ>
```
ebXML Metadata and Items

Two basic types:
- Core Components – (nouns)
- Business Processes – (verbs)
This is over simplified
Core Components are:

- A Core Component captures information about a real world (business) concept, and relationships between that concept and other business concepts.

- A Core Component can be either an individual piece of business information, or a natural 'go-together' family of business information pieces.

- It is ‘Core’ because it occurs in many different areas of industry/business information exchange.
ebXML Core Components

- Vocabularies (eg. xCBL 2.0) contain elements that may be semantically identical to some of the common core components. Examples can be an `<address>` element on a xCBL invoice and the `<partyAddress>` on a Visa XML Invoice.
- Core Components must have contextual identity at run time
  - i.e. `PurchaseOrder.sendParty(name) != PurchaseOrder.sendParty(name)`
ebXML Business Process

- BP describe document choreography and overall process interfaces.
- Identify which data needs to be present to ensure requirements of both parties are being met.
- Examples can be “Deliver a service” or “Purchase a product”
Repository Item Examples

• XML elements in business messages can reference items in a repository.

• Examples:
  – <nameofperson>
  – <nomdelapersonne>
  – <name>
  – <TheThingICallYou>

  All are the same item (semantically)!!!
The Process Layer Example

• Examples:
  – <name_of_person>
  – <nom_du_personne>
  – <name>
  – <TheThingICallYou>

• Are all XML elements that may be part of a process called “Purchase a Product”
The Layers operate in Parallel

- XML Elements:
  - `<name_of_person GUID="12345">
  - `<nom_du_personne GUID="12345">
  - `<name GUID="12345">

- Process
  - `<Process GUID="678">Purchase a Product</Process>
  - `<Step_1>
    - `<Item sentBy="party_1" to="party_2" GUID="12345">name</Item>`
How ebXML Trading Partners interact

- A Trading Partner can create a model of its business and business items. Isn’t always necessary – ie. SME’s can buy packages from ASP’s which will likely use existing vocabularies (xCBL, cXML, Visa XML et al).
- A Trading Partner can also identify and use components/processes used by its partners.
How ebXML Trading Partners Interact

- The Trading Partner sends a business message instance to another ebXML capable trading partner.
- The business message is part of a Business Process
How ebXML Trading Partners Interact (2)

- The transaction contains abstractions of two layers – the Core Components (noun) layer and the Business Process (verb) layer.

![Diagram showing the interaction between Trading Partners and repositories]

Trading Partner #1 (ebXML Compliant) — Business Interchange — Trading Partner #2 (ebXML Compliant)

- CC Repository
- BP Repository
- Contains References to
- Nouns
- Verbs
Business Message References

- Partner Discovery Layer
  - Examples: eCo.xml

- Business Process Layer
  - Examples: ebXML BP Syntax

- Business Info Layer
  - Examples: xCBL, cXML Visa Inv.

- Business Messages
  - Contains

- References to:
  - Human Search Interface
  - Business Application Interface
  - API
  - Registry
  - Repository Items

- Human Actors
<xml version="1.0"?>
<purchase_a_product GUID="678">
<Name GUID="12345">Duane</Name>
...
</purchase_a_product>
1. Request ebXML specification
2. Send ebXML specification
3. Request to upload company information
Profile & Scenarios updated OK

DO BUSINESS!

Query about Company X
Send Company X’s Profile
Request Company X’s Scenario
Send Company X’s Scenario

ebXML BO Library

Company X

Company Y

ebXML Repository
Profiles
Scenarios
Some Final Thoughts..

- ebXML to build an open architecture, not a “Standard”
- Truly interoperable and Extensible (Global)
- Includes everyone from SME’s to Fortune 1000

Thank you!

Duane Nickull