Why Are We Talking About UBL

- UBL fulfils the promise of XML for business by defining a standard cross-industry vocabulary
- UBL is the ebXML missing link
- UBL plus ebXML enables the next generation of eBusiness exchanges
  - Cheaper, easier, Internet-ready
  - Extends benefits of EDI to small businesses
  - Fits existing legal and trade concepts
  - Allows re-use of data
- UBL can provide the XML payload for a wide variety of other web-based business frameworks
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary

Toward a Universal Business Language
The promise of XML for e-business

• Plug ‘n’ play electronic commerce
  – Spontaneous trade
  – No custom programming

• Ubiquity on the Internet
  – Dirt-cheap tools
  – Complete platform independence
  – Enable true global market availability

• Enable universal interoperability
  – Abandon existing EDI systems
  – Handle both "publication" document types and "transactional" documents
Goals for Successful eBusiness Services

- Web-enable existing fax- and paper-based business practices
- Allow businesses to upgrade at their own pace
- Preserve the existing investment in electronic business exchanges
- Integrate small and medium-size businesses into existing electronic data exchange-based supply chains

The standardization of XML business documents is the easiest way to accomplish these goals.
XML For Interoperability???

- anyML
- anyML

Differences in:
- tags & meanings
- positions
- informational content

Different information elements

TOWARD A UNIVERSAL
Can't We Just Do It?

TOWARD A UNIVERSAL BUSINESS LANGUAGE
The Role of a Hub Format

One adapter interfaces all suppliers to a common consumer...

as well as all consumers to a common supplier... and all businesses to the tax authorities and the customs agents and the accountants and the transporters ...

There appears to be no practical alternative to this plan.

Toward a Universal Business Language
Why UBL?

• Leverages knowledge from existing EDI and XML B2B systems
• Applies across all industry sectors and domains of electronic trade (80/20 rule)
• Fully leverages XML
• Provides the ebXML missing link:
  - XML Payload +ebXML CCTS +modelling methodology
• Non-proprietary and committed to freedom from royalties
• On track to become international standard for trade through ISO TC154
UBL and the ebWS stack

UBL Formatting Specs
UBL Context Methodology
UBL Library
Core Components
ebXML CPPA
ebXML BPSS
ebMS

Message presentation
Message contextualization
Standard messages
Business agreements
Business processes
Packaging/transport

Toward a Universal Business Language
ebXML Core Components

• A set of the lowest common denominator that captures information about a real world (business) concept

• Core Components are neutral
  – in the notation for every kind of industry
  – in the syntax for every kind of business document standard or implementation
ebXML Core Components

• Reusable pieces (objects) of contents that can be atomic or aggregate
  – Enables interoperability among different industry domains and areas
  – Are using common semantic units at any level consistent across context
  – Hold any related information together and avoiding fragmented semantic dispersal
  – Facilitate multilingual support

• Accompanied by methodology for extensibility
  – Enable users to define meaningful business and process data
  – Ensure maximum interoperability
The Core Components Specification Follows
ISO 11179

Object class

- Property 1: representation 1
- Property 2: representation 2
- Property 3: representation 3
- Property 4: representation 4

Address

- Street: text
- Post code: text
- Town: text
- Country: identifier

ISO 11179 governs data dictionaries:
defines the notions of object class, property, and representation term

- This is basic object-oriented “good stuff”

Toward a Universal Business Language
The ebXML CCTS system

- **Core Component Type (CCT)**
  - **Content Component**: Consists of
  - **Supplementary Component**: Consists of

**Data Type**

- **Basic Core Component**: Defines set of values of
- **Association Core Component**: Provides a simple characteristic of and is aggregated in
- **Aggregate Core Component**: Provides a complex characteristic of and is aggregated in

Without business semantics

With known business semantics

TOWARD A UNIVERSAL BUSINESS LANGUAGE
Applying Business Context

Core

Core Component Type (CCT)

Data Type

Basic Core Component

Association Core Component

Aggregate Core Component

Business

Data Type

Basic Business Information Entity

Association Business Information Entity

Aggregate Business Information Entity

Assembly Component

Message Assembly

Core Component Library
Core Components vs. Business Information Entities

Core Component (CC)

A building block for the exchange of semantically correct and meaningful information

Business Information Entity (BIE)

A CC to which a business context has been applied

apply business context:
- business process
- product classification
- industry classification
- geopolitical region
- official constraint
- business process role
- supporting role
- system capabilities

• An address might be a generic CC
• A U.S. address has (at least) the geopolitical region set as its business context, making it a BIE
• UBL, by its nature, deals only in BIEs
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary
NDR Requirements

• Leverage XML technology, but keep it interoperable

• Achieve semantic clarity through a binding to the Core Components model

• Support contextualization (customization) and reuse

• Selectively allow “outsourcing” to other standard schemas
Mapping CCTS to XSD
Some Major Design Rules

• The choice of normative schema language - XSD

• Garden of Eden design approach
  – Named Types
  – Global Elements for all aggregate and leaf elements except:
    • Local Elements for Identifiers and Codes

• Naming and construction of elements, attributes, and types

• Modularity, namespaces, and versioning

• Embedded schema documentation

• Handling code lists (preliminary)
A Taste Of The Naming Rules

• Dictionary entry names are fully qualified with object class names
• But using these full names would result in hundreds of extra elements
• We get reusability by allowing properties (elements) to “inherit” parent object classes (types), XPath-style
  – Delivery schedule IDs and order IDs could both be called <ID>
  – Each would be identifiable by means of //Order/ID and //DeliverySchedule/ID respectively
• Use of abbreviations and acronyms is severely limited
UBL Namespaces

- **File**
- **W3C XML Schema**
- **Namespace**

**ControlSchema**

- **InternalSchemaModule**
  - included
  - 0..*

- **ExternalSchemaModule**

  - imported
  - 4..*

Shaded area is a "schema set".

The four required namespaces are (represented by their prefixes): `dt`, `rt`, `cbc`, `cac`.

{In same namespace as ControlSchema}

{In different namespace than ControlSchema}
CBC: Common Basic Components
CAC: Common Aggregate Components
DT: DataTypes
RT: RepresentationTerms (not CodeTypes)
CTRT: CodeTypeRepresentationTerms

CBC, CAC, DT, RT, and CodeList schema modules are imported by ControlSchema. CTRT is not.
Reusing Components Across Control Schemas
Code Lists

• UBL will seek to import external datatype definitions in conventional XSD form
  – Validation
  – Clarity

• We are developing a schema for promotion as an international standard

• We hope to promote a global code list marketplace
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary

Toward a Universal Business Language
Document Engineering

- A new discipline for designing electronic document structures
- Applicable to Internet information exchange mechanisms to request or return the results of business processes (aka Web Services).
- Consolidates document and data-centric perspectives
- Provides analysis and design methods that yield formal models
- UBL has developed these models for a common business process – procurement.
Business Operations View

UML and spreadsheets

Conceptual models showing all possible associations

Analysis

XML Schema Libraries

Database Schemas

EDI Message Definitions

Business Document Models

Functional

Service view

Schemas

The Real

World

Messages/Documents

Limited interoperability

UBL Analysis

Toward a Universal Business Language
A Conceptual 'Item'

Toward a Universal Business Language
Designing Document Models

- Create document model(s)
  - structures are assembled from conceptual model components into hierarchies
  - pathways based on context/business rules

- Implementing document model(s)
  - assembled document tree models can easily map to schema languages
  - UBL automates this for XML XSD

Toward a Universal Business Language
Business Operations View

UML and spreadsheets

Design

Functional Service view

Schemas

Analysis

XML Schema Libraries

EDT Message Definitions

Business Document Models

Documents structures are assembled from 'network' of components into document models

Limited interoperability

UBL Design

TOWARD a UNIVERSAL BUSINESS LANGUAGE
<table>
<thead>
<tr>
<th>UBL Name</th>
<th>BIE Dictionary Entry Name</th>
<th>Occurrence</th>
<th>BIE Type</th>
<th>UBL Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Item. Description. Text</td>
<td>0..1</td>
<td>BBIE</td>
<td>a free form field that can be used to give a text</td>
</tr>
<tr>
<td>PackQuantity</td>
<td>Item. Pack. Quantity</td>
<td>0..1</td>
<td>BBIE</td>
<td>the unit packaging quantity.</td>
</tr>
<tr>
<td>PackSizeQuantity</td>
<td>Item. Pack_Size. Quantity</td>
<td>0..1</td>
<td>BBIE</td>
<td>the number of items in a pack.</td>
</tr>
<tr>
<td>FromCatalogueIndicator</td>
<td>Item. From Catalogue. Indicator</td>
<td>0..1</td>
<td>BBIE</td>
<td>an indicator that denotes whether or not the item was</td>
</tr>
<tr>
<td>BuyersItemIdentification</td>
<td>Item. Buyers_Item Identification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its identification according to the buyers system.</td>
</tr>
<tr>
<td>SellerItemIdentification</td>
<td>Item. Sellers_Item Identification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its identification according to the sellers system.</td>
</tr>
<tr>
<td>ManufacturersItemIdentification</td>
<td>Item. Manufacturers_Item Identification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its identification according to the manufacturers system.</td>
</tr>
<tr>
<td>StandardItemIdentification</td>
<td>Item. Standard_Item Identification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its identification according to a standard system.</td>
</tr>
<tr>
<td>CatalogueItemIdentification</td>
<td>Item. Catalogue_Item Identification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its identification according to a cataloging system.</td>
</tr>
<tr>
<td>AdditionalItemIdentification</td>
<td>Item. Additional_Item Identification</td>
<td>0..n</td>
<td>ASBI</td>
<td>associates the item with other identification means</td>
</tr>
<tr>
<td>CatalogueReference</td>
<td>Item. Catalogue_Reference</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with the catalogue from which the item was selected.</td>
</tr>
<tr>
<td>OriginCountry</td>
<td>Item. Origin_Country</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its country of origin.</td>
</tr>
<tr>
<td>CommodityClassification</td>
<td>Item. Commodity Classification</td>
<td>0..1</td>
<td>ASBI</td>
<td>associates the item with its classification(s) according to a commodity classifying system.</td>
</tr>
<tr>
<td>SalesConditions</td>
<td>Item. Sales Conditions</td>
<td>0..n</td>
<td>ASBI</td>
<td>associates the item with sales conditions appertaining to it.</td>
</tr>
<tr>
<td>HazardousItem</td>
<td>Item. Hazardous Item</td>
<td>0..n</td>
<td>ASBI</td>
<td>associates the item with its hazardous item information.</td>
</tr>
<tr>
<td>TaxCategory</td>
<td>Item. Tax Category</td>
<td>0..n</td>
<td>ASBI</td>
<td>associates the item with one or more taxes</td>
</tr>
<tr>
<td>BasePrice</td>
<td>Item. Base Price</td>
<td>0..n</td>
<td>ASBI</td>
<td>associates the item with one or more base prices.</td>
</tr>
</tbody>
</table>
Business Operations View

UML and spreadsheets

Design

UBL script does this automatically

Service View

Schemas

Analysis

XML Schema Libraries

Database Schemas

EDI Message Definitions

Business Document Models

The Real World

Messages/Documents

Limited interoperability

UBL Encoding

Toward a Universal Business Language
EbXML CC names to XML tag names

- Remove redundant and nearly redundant words in the property field (as in *. Identification. Identifier)
- Remove periods, spaces, and underscores
- When the representation term is “Text”, remove it
- Remove the object class name on properties, as the XML parent labels it sufficiently
- These are applied automatically by the schema generator

Item. Details
Item. Description. Text
Item. Pack. Quantity
Item. Buyers_ Item Identification
Item. Sellers_ Item Identification

ItemType
Description
PackQuantity
BuyersItemIdentification
SellersItemIdentification
<xsd:complexType name="ItemType">
    <xsd:annotation>
        <xsd:documentation>
            <ccts:Component>
                <ccts:CategoryCode>ABIE</ccts:CategoryCode>
                <ccts:DictionaryEntryName>Item. Details</ccts:DictionaryEntryName>
                <ccts:Definition>Information directly relating to an item</ccts:Definition>
                <ccts:ObjectClass>Item</ccts:ObjectClass>
                <ccts:PropertyTerm>Details</ccts:PropertyTerm>
                <ccts:RepresentationTerm>Details</ccts:RepresentationTerm>
                <ccts:BusinessTerm>article, product, goods item</ccts:BusinessTerm>
            </ccts:Component>
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element ref="Description" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
Business Operations view

UML and spreadsheets

Design

Functional

Final presentation is application dependent (e.g. Stylesheet)

The Real World

Messages/Documents

Limited interoperability

Greater interoperability

UBL Implementation

Toward a Universal Business Language
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary
Customizing Means

• Applying a different context to the use of the schemas, e.g.
  – Adding Japanese accounting requirements to the UBL 'Payment' structure when used in Invoice.
  – Re-using UBL components

• Using UBL with other vocabularies, e.g.
  – RosettaNet, OAG
  – EDIFACT, ANSI X12
  – Proprietary formats
Levels of Interoperability

- **Schema compatibility**
  - My schema for UBL Invoice still parses the Japanese one
  - UBL Context Methodology

- **Syntax compatibility**
  - I need your schema but my application understands the structure used
  - We share the same NDRs for XML

- **Semantic compatibility**
  - Whatever you call the component I can understand what it means and can map it to my application
  - We share ebXML “syntax-independent” core components
Business Operations view

UML and spreadsheets

Design

Syntax Level

Functional

Semantic Level

The Real

World

Schemas

Messages/Documents

XML Schema

Libraries

EDI Message

Definitions

Business

Document

Models

Limited interoperability

Levels of Re-use

Toward a Universal Business Language

1. Plan for Today's Class

• What is a 'model'?
• What is a 'metamodel'?
• Why do modeling?
• Disciplines for modeling
• Just enough UML
• XML as a modeling notation

BuyerParty

Party

NameAddress

Name1

ABC Enterprises

Name2

Global Services

POBox

POBoxPostalCode=

PostalCode

20012

City

Alpine

Region

RegionCoded

USNY

Implementation

Analysis

Schema Level

XML Schema

Libraries

EDI Message

Definitions

Business

Document

Models
Context Methodology

- Defines how document formats can be extended based on specific trading partner characteristics
- Takes ebXML context drivers (8 space) and context rules as starting point:
  - Business Process, Product Classification, Industry Classification, Geopolitical, Official Constraints, Business Process Role, Supporting Role, System Capabilities
- Builds on experience with OO extension methodology, but will be
  - More structured
  - More consistent
  - Easier to track
  - Easier to automate
  - Require a lower level of skill
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary

TOWARD A UNIVERSAL BUSINESS LANGUAGE
Normative Schemas

- W3C XML Schema (XSD) modules
  - Schema of re-usable Types
    - Party, Address, Item, Tax, etc
  - Schemas for ebXML Core Component Types
    - Core Component Types, Representation Terms and Data Types
  - Schemas for documents:
    - Order, Order Change, Order Cancellation, Order Response(simple), Order Response(complex), Despatch Advice, Receipt Advice, Invoice
  - Schemas for Code Lists (enumerations)
    - For each UBL code type
    - Currently all “placebo”s
Release Schema Structure

The control document schema in...
xsd/maindoc/UBL-Order-1.0-beta.xsd

Refers to the common UBL Library in...
xsd/common/UBL-Reusable-1.0-beta.xsd

That uses data types defined in...
xsd/common/UBL-RepresentationTerms-1.0-beta.xsd

That extends data types defined in...
xsd/common/UBL-CoreComponentTypes-1.0-beta.xsd

Any of these may refer to a list of code values defined in...
xsd/codelist/use/UBL-CodeList-DocumentStatusCode-Use-1.0-beta.xsd
Normative Business Process

• Tied to document types used in schemas
  – Basic Order-to-Invoice
  – Defines business rules/constraints for document use
  – 80/20 rule applies
Non-Normative Data Models

• **Conceptual Model**
  - UML (design format)
  - Party, Address, Item, Tax, etc...
  - "Candidate" ebXML Core Components

• **Document Models**
  - Spreadsheets (maintenance format)
  - Assembled from conceptual model components (Re-usable Types)
  - Order, Order Change, Order Cancellation, Order Response(simple), Order Response(complex), Despatch Advice, Receipt Advice, Invoice
  - Apply UBL naming rules by formulae
Other Non-Normative Deliverables

- Checklist of UBL naming and design rules
- Tool(s) for generating the schemas
- Additional modelling views:
  - Implementation view UML (actual schema structure)
  - ASN.1 schema
  - RELAX NG schema(?)
- Forms presentation mappings for developing style sheets, etc.
- Sample XML instances and outputs
- Usage documentation
Package Availability

• HTML, PDF and OpenOffice document formats

Available at:
http://www.oasis-open.org/committees/ubl/lcsc/UBLv1-beta
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary

Toward a Universal Business Language
UBL Status

- November 2003
  - UBL 1.0 beta (Committee Draft) for pilot implementation
  - Feedback essential to prove the specification

- January 2004
  - UBL NDR 1.0 Committee Draft

- February 2004
  - UBL 1.0 & UBL NDR 1.0 Committee Draft to OASIS for approval

- March/April 2004
  - UBL 1.0 & UBL NDR 1.0 become OASIS Technical Specification
  - UBL 1.0 and UBL NDR 1.0 Fast Tracked to ISO TC154

- UBL is at a watershed
  - 1.0 Library is complete(almost!)
  - Planning for UBL 2.0 starts now!
  - Customizing UBL for other contexts
UBL Technical Subcommittees

- **NDRSC**
  - **Naming and Design Rules SC**
  - Develops guidelines for normative-form schema design, instance design, and markup naming.

- **CMSC**
  - **Context Methodology SC**
  - Develops a methodology and tools for applying context.

- **CDSC**
  - **Context Drivers SC**
  - Works on improvement and further development of the context drivers.

- **TTSC**
  - **Tools and Techniques SC**
  - Evaluates and recommends the tools and techniques for development, maintenance and revision.

- **LCSC**
  - **Library Content SC**
  - Defines business documents and a library of XML and ebXML CCTS based building blocks.

- **FPSC**
  - **Forms Presentation SC**
  - Develops and documents formal technology-agnostic Formatting Specifications.

- **JPLSC**
  - **Japanese Localization SC**
  - Customization/Implementation of UBL in the Japanese market.

- **CNLSC**
  - **Chinese Localization SC**
  - Customization/Implementation of UBL in the Chinese market.

- **CLSC**
  - **Code List SC**
  - Work with NDR/LC to finalize and institutionalize UBL Code List solution.
UBL Technical Subcommittees

- NDRSC: Naming and Design Rules SC
  Develops guidelines for normative-form schema design, instance design, and markup naming.

- CMSC: Context Methodology SC
  Develops a methodology and tools for applying context.

- CDSC: Context Drivers SC
  Works on improvement and further development of the context drivers.

- TTSC: Tools and Techniques SC
  Evaluates and recommends the tools and techniques for development, maintenance, and revision.

- LCSC: Library Content SC
  Defines business documents and a library of XML and ebXML CCTS based building blocks.

- FPSC: Forms Presentation SC
  Develops and documents formal technology-agnostic Formatting Specifications.

- JPLSC: Japanese Localization SC
  Customization/Implementation of UBL in the Japanese market.

- CNLSC: Chinese Localization SC
  Customization/Implementation of UBL in the Chinese market.

- CLSC: Liaison SC
  Work with NDR/LC to finalize and institutionalize UBL Code List solution.
UBL Technical Subcommittees

NDRSC
Naming and Design Rules SC
Develops guidelines for normative-form schema design, and more.

CMSC
Context Methodology SC
Develops a methodology and tools for applying context.

CDSC
Context Drivers SC
Works on improvement and further development of the context drivers.

Tools and Techniques SC
Evaluates and recommends the tools and techniques for development, maintenance and revision.

NDRSC
Naming and Design Rules SC
Develops guidelines for normative-form schema design, and more.

Implementation SC
- Monitor UBL-Dev and coordinate with other technical SC’s for response
- Collate issues discovered and convey them to the appropriate SC for action

JPLSC
Japanese Localization SC
Customization/Implementation of UBL in the Japanese market

CNLSC
Chinese Localization SC
Customization/Implementation of UBL in the Chinese market

CLSC
Liaison SC
Work with NDR/LC to finalize and institutionalize UBL Code List solution

Toward a Universal Business Language
UBL Administrative Subcommittees

ASC
Administration SC
Administrates and coordinates the UBL efforts

MSC
Marketing SC
Does marketing and promotion for the UBL effort

LSC
Liaison SC
Organizes liaisons with other organizations.
Some UBL Participants

APACs
Boeing
Commerce One
Danish Bankers Association
France Telecom
General Electric
Government of Hong Kong
Government of Korea
HP
Intuit
KPMG
LMI

Northrup Grumman
Oracle
PricewaterhouseCoopers
SAP
SeeBeyond
Sterling Commerce
Sun Microsystems
UCB Center for Document Engineering
UK Cabinet Office
United Parcel Service
U.S. GSA
U.S. Navy
Visa International

Toward a Universal Business Language
The Value of Joining Forces

• As a non-profit cross-industry effort, UBL depends on expert domain input to “get it right”

• We actively solicit industry and standards liaisons

• Organizations appoint representatives to the UBL Liaison Subcommittee
  – If the organization is not an OASIS member, an individual representative joins at USD 250/year
  – Telcons are held frequently
  – Liaisons arrange for specification reviews
Formal Liaisons (so far)

- ACORD (insurance)
- ARTS (retail sales)
- ebXML Asia Committee (ebXML)
- e.centre (EAN UK)
- EIDX (electronics)
- HL7 (healthcare)
- Information Technology Standards Committee of Singapore
- NACS (convenience stores)
- Open Applications Group
- RosettaNet (information technology)
- SWIFT (banking)
- UIG (utilities)
- VCA (optical supplies)
- XBRL (accounting)
- ASC X12 COTG
- UN/CEFACT TBG
- UN/CEFACT ATG
- OASIS eGov TC
- OASIS CIQ TC
Basic UBL Documents

• Procurement
  – Purchase Order, P.O. Response, P.O. Change

• Materials management
  – Advance Ship Notice, Planning Schedule, Goods Receipt

• Payment
  – Commercial Invoice, Remittance Advice

• Transport/logistics
  – Consignment Status Request, Consignment Status Report, Bill of Lading

• Catalogs
  – Price Catalog, Product Catalog

• Statistical reports
  – Accounting Report
Expand The Library

• Fill in basic list
• Add new work areas
  – Tax
  – Customs
  – Government
  – Other
Work the Customization Methodology

• Move forward with Context Methodology
• Develop routines to automatically apply context and auto-generate UBL conformant contextualized schemas
Overview

1. What and Why of UBL
2. The Design of UBL
   - ebXML Core Components
   - Naming and Design Rules
   - Document Engineering
   - Customizing UBL
3. The Content of UBL 1.0
   - What is Normative
   - What is non-Normative
   - Availability
4. Making UBL Happen
5. UBL Phase 2
6. Summary

Toward a Universal Business Language
UBL Offers Important and Interesting Solutions

• As a B2B standard:
  – It is user-driven, with deep experience and partnership resources to call on
  – It is committed to truly global trade and interoperability
  – Its process is open and transparent

• As a Web Service interface:
  – It can be layered on existing standards
  – It is providing the 'missing link'; content

• As an e-business vocabulary:
  – 80/20 plus customisation
Where To Find More Information

• OASIS UBL TC
  – White papers, presentations, and specifications are available
  – All mailing list archives are open to public view

• For ebXML: [http://www.ebxml.org](http://www.ebxml.org)

• For ebXML Core Components:
  <chair: Alan.Stitzer@marsh.com>
  <editor: mcrawford@lmi.org>

• For LMI’s work for US Government Agencies: [http://www.lmi.org](http://www.lmi.org)

• For LMI’s XML work: <mcrawford@lmi.org>
Thanks! Questions?