COMPOSING MEANINGFUL DOCUMENTS WITH UBL

Tim McGrath
Open Standards 2008: Composability within SOA Symposium
Date: 28 April – 1 May
Location: Santa Clara Marriott
- an approach to design and customization of UBL to satisfy specific requirements

UBL is establishing itself as the language of choice for a variety of electronic commerce applications. It is particularly well suited for use in business web services and service oriented architectures. The use of UBL for such environments ensures that services remain meaningful and worthy in a variety of different compositions. Inevitably, these applications may require some refinement of the standard structures.

This presentation explains some of the design principles behind UBL and how both the standard components and the principles behind them can be adapted to suit the requirements of different business processes. We will explore a case study involving the Danish Government's OIOUBL and Service Oriented Infrastructure project.

Composing different services into a meaningful business process requires interoperability of the information contained in the documents exchanged. This tutorial will discuss the opportunities for utilizing the OASIS Universal Business Language (UBL) as the glue to bind these services together at a business data level.

This tutorial explains the design principles behind UBL and how it can be adapted to suit the requirements of different business processes. These composite applications will require some customization of the standard structures and the creation of new document types.

The objective is to enable participants to understand the issues involved in customizing UBL and how to develop specific implementations that suit the requirements of their context of use. The target audience is system analysts and developers responsible for the implementation of document interfaces.
Outline

- Background to Document Design
- UBL Design
- UBL Specification
- Case Study Scenario
- Designing Customizations
- Specifying Customizations
- Validating Customizations
- Summary and Feedback
Introductions
Who are we and why are we here?
Background to Document Design
What are Documents?

• For nearly three thousand years information has been organized in purposeful and self-contained packages.
  - We call them documents.

• Using documents for exchanging business information is an old idea.
  - The technology for encoding and exchanging documents has profoundly changed
  but
  - The concept of a document has remained surprisingly stable.
Using documents for exchanging business information is natural and intuitive.

Administrative documents
Early Dynastic III, about 2500 BC
Probably from Shuruppak (Fara)

Left: the governor receives 14 shekels of silver from 5 persons, and 46 shekels remain unpaid by 13 other persons. The total amount of silver is 1 mina.
<cac:TaxTotal>
  <cbc:TaxAmount currencyID="SHEKEL">60.00</cbc:TaxAmount>
  <cac:TaxSubtotal>
    <cbc:TaxableAmount currencyID="SHEKEL">100.00</cbc:TaxableAmount>
    <cbc:TaxAmount currencyID="SHEKEL">14.00</cbc:TaxAmount>
  </cac:TaxSubtotal>
</cac:TaxTotal>

<cac:TaxCategory>
  <cac:TaxScheme>
    <cbc:ID>ASSYRIAN GOVERNOR TAX</cbc:ID>
  </cac:TaxScheme>
</cac:TaxCategory>
</cac:TaxSubtotal>
</cac:TaxTotal>
How Documents Work

• Documents are self-contained collections of information.
  - Interfaces for people.
  - Interfaces to business processes.
• To be meaningful documents must be defined using a common language.
  - Not for information content but
  - For describing the meaning (semantics) of the information content.
• The Universal Business Language is a semantic document description language.
Document Exchange

• A document exchange consists of both processes and the documents (information) they produce and consume.

• By understanding the processes
  - we learn what kinds of information is needed.

• By understanding the document
  - we learn what kinds of processes are possible.
Interoperability

• A basic requirement for two businesses to conduct business is that their business systems interoperate.
• Easy to express but hard to achieve.
• Variations in strategies, technology platforms, legacy applications, business processes, and terminology.
• Interoperability doesn’t require that two systems be identical.
  - Just they have common understanding of the meanings of things (semantics)
UBL Design
UBL Design

ebXML Core Components
• **ebXML Core Components Technical Specification:**
  - ISO 15000-5.
  - UN/CEFACT

• How to describe core components of business documents in a syntax independent manner.

• UBL is the first set of integrated XML document specifications based on CCTS.
  - UN/CEFACT recognizes UBL as appropriate first-generation XML documents for eBusiness.
CCTS Principles

These are implied concepts

Core

Aggregate Core Component (BCC)

Basic Core Component (BCC)

is of type

Core Component Type (CCT)

is defined in context as

Business

Aggregate Business Information Entity (ABIE)

Basic Business Information Entity (BBIE)

is defined in context as

UBL publishes these

is defined in context as

Message/Document

These are implied concepts
Business Information Entities

- Reusable building blocks for the exchange of information ("Core Components") in a specific Business Context.
- **Aggregate BIE**
  - A collection of related pieces of business information that together convey a distinct business meaning.
- **Association BIE**
  - Is associated to an Aggregate Business Information Entity, which describes its structure.
- **Basic BIE**
  - A singular business characteristic of a specific aggregation.
Basic Business Information Entities

- A singular business characteristic in a specific business context.
- Individual pieces of information.

```
<ABIE>
An Address

# «BBIE» Street Name: Name [0..1]
# «BBIE» City Name: Name [0..1]
```
Aggregate Business Information Entities

- A collection of related pieces of business information.
- A container of BBIEs and ASBIEs to other ABIEs.

```
<ABIE>
An Address

# <BBIE> Street Name: Name [0..1]
# <BBIE> City Name: Name [0..1]
```
Association Business Information Entities

- Represents an ABIE in a specific business context.
- An ABIE contained within another ABIE.
UBL Design

Naming Information Entities
Names

• Meaningful Information Entity names.
  - promote a common understanding.
  - encourage reuse.

• May need different names in different modeling artifacts.
  - XML Tag may not be the modeling or the presentation name.
Choosing the Right Names

- Selecting meaningful terminology for names is a craft not a science.
  - Think about spoken languages.
  - No two designers will choose the same terms.
- Levels of discipline:
  - Dictionaries.
  - Controlled vocabularies.
  - Formal ontologies.
- We need more information than just names.
- We also need to understand the context of use.
Rules for Consistent Naming

- Names should be grammatically consistent.
  - Singular not plurals
- Names should not be ambiguous.
  - Is “Supplier’s ID” the same as “ID of Supplier”?
- Check names are not semantically misleading.
  - “Product Number” may not be a number, so “Product Identifier” is better.
- Qualify names to clarify the context of use.
  - “Customer Party” is more meaningful than “Customer”.
ebXML Naming Rules

- ebXML Core Components naming rules are an application of ISO 11179.
- All Information Entities get a Dictionary Entry Name (DEN).
- Dictionary Entry Names are globally unique.
- They are the ‘key’ or unique identifier for any Information Entity.
Dictionary Entry Names

\{\text{qualifier}\} \textbf{Object Class Term}, ". ", \{\text{qualifier}\} \textbf{Property Term}, ". ", \{\text{qualifier}\} \textbf{Representation Term}

- **Object Class Term**
  - The object class to which it belongs.

- **Property Term**
  - Reflecting its function as a property or distinguishing characteristic of the object class.

- **Representation Term**
  - How it is represented.
Examples of CCTS Names

- The name for ‘The currency in which the tax is collected and reported, expressed as a code’:


↑  Object Class  Property  Representation

ABIE  BBIE  Data Type
Qualified CCTS Names

- The name for ‘An indicator as to whether these totals are recognized as legal evidence for taxation purposes’:

Tax Total. Tax Evidence_ Indicator. Indicator

Object Class

Qualifier

Property

Representation

Data Type
CCTS Truncation Rule

- The name for ‘Identifies the tax scheme’:

  Tax Scheme. Identifier. Identifier

- Can be truncated to remove duplicate terms:

  Tax Scheme. Identifier
UBL Design

Representation Terms
Amount

A number of monetary units specified in a currency where the unit of currency is explicit or implied.

The currency of the amount.

The version of the UN/ECE rec. 9 code list.

<cbc:TaxAmount currencyID="DKK"> 1262.50 </cbc:TaxAmount>
<table>
<thead>
<tr>
<th>Binary Object. Content</th>
<th>A set of finite-length sequences of binary octets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Object. Format.</td>
<td>The format of the binary content.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Binary Object. Character Set. Code</td>
<td>The character set of the binary object if the mime type is text.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Binary Object. Uniform Resource. Identifier</td>
<td>The Uniform Resource Identifier that identifies where the binary object is located.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>Binary Object. Filename. Text</td>
<td>The filename of the binary object.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<cbc:EmbeddedDocumentBinaryObject mimeCode="application/pdf" encodingCode="Base64" characterSetCode="UTF-8">
brlG0Ng53KHKaoOQNQYcpyFHKa9UhkgzjkPBTaFJDIBUasfdgartujuoprrwer34578deewcwetyuuutyur643432THyB4KY5SYM5BZcpMkshU0UzJoLHSDEWFdreefssd///<
</cbc:EmbeddedDocumentBinaryObject>
```
Code

A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute.

The identification of a list of codes.

An agency that maintains one or more code lists.

The name of the agency that maintains the code list.

<cbc:ItemClassificationCode listID="UNSPSC" listVersionID="7.0401">43191501</cbc:ItemClassificationCode>
<table>
<thead>
<tr>
<th>Code List. Name. Text</th>
<th>The name of a list of codes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code List. Version.</td>
<td>The version of the code list.</td>
</tr>
<tr>
<td>Identifier</td>
<td></td>
</tr>
<tr>
<td>Code. Name. Text</td>
<td>The textual equivalent of the code content.</td>
</tr>
<tr>
<td>Language. Identifier</td>
<td>The identifier of the language used in the corresponding text string.</td>
</tr>
<tr>
<td>Code List. Uniform</td>
<td>The Uniform Resource Identifier that identifies where the code list is located.</td>
</tr>
<tr>
<td>Resource. Identifier</td>
<td></td>
</tr>
<tr>
<td>Code List Scheme.</td>
<td></td>
</tr>
<tr>
<td>Uniform Resource.</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<cbc:ItemClassificationCode listID="UNSPSC" listVersionID="7.0401">43191501</cbc:ItemClassificationCode>
```
Date Time
(Date, Time)

Date Time. Content

The particular point in the progression of time.

Date Time. Format. Text

The format of the date/time content.

<cbc:StartDate>2007-01-25</cbc:StartDate>
Identifier

A character string used to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme.

The identification of the identification scheme.

The name of the identification scheme.

<cbc:ID schemeID="GTIN">5712345780121</cbc:ID>
Identifier (contd.)

<table>
<thead>
<tr>
<th>Identification Scheme</th>
<th>The identification of the agency that maintains the identification scheme.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agency. Identifier</td>
<td>The name of the agency that maintains the identification scheme.</td>
</tr>
<tr>
<td>Identification Scheme.</td>
<td>The version of the identification scheme.</td>
</tr>
<tr>
<td>Agency Name. Text</td>
<td>The Uniform Resource Identifier that identifies where the identification scheme data is located.</td>
</tr>
<tr>
<td>Identification Scheme.</td>
<td>The Uniform Resource Identifier that identifies where the identification scheme is located.</td>
</tr>
<tr>
<td>Version. Identifier</td>
<td></td>
</tr>
<tr>
<td>Identification Scheme.</td>
<td></td>
</tr>
<tr>
<td>Data. Uniform Resource. Identifier</td>
<td></td>
</tr>
</tbody>
</table>
Indicator

**Indicator. Format.**
- Text

**Indicator. Content**
- The value of the indicator.

Whether the indicator is numeric, textual or binary.

```
<cbc:ChargeIndicator>false</cbc:ChargeIndicator>
```
<table>
<thead>
<tr>
<th>Measure. Content</th>
<th>The numeric value determined by measuring an object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure Unit Code</td>
<td>The type of unit of measure.</td>
</tr>
<tr>
<td>Measure Unit Code</td>
<td>The version of the measure unit code list.</td>
</tr>
<tr>
<td>List Version. Identifier</td>
<td></td>
</tr>
</tbody>
</table>

```
<cbc:GrossWeightMeasure unitCode="KGS"> 21408.79 </cbc:GrossWeightMeasure>
```
Numeric
(Value, Rate, Percent)

**Numeric Content**

Numeric information that is assigned or is determined by calculation, counting or sequencing.

**Numeric Format**

Whether the number is an integer, decimal, real number or percentage.

```xml
<cbc:PackSizeNumeric>1</cbc:PackSizeNumeric>
```
Quantity

- **Quantity. Content**: A counted number of non-monetary units possibly including fractions.
- **Quantity. Unit. Code**: The unit of the quantity.
- **Quantity Unit. Code List Agency. Identifier**: The quantity unit code list.
- **Quantity Unit. Code List Agency Name. Text**: The identification of the agency that maintains the quantity unit code list.
- **Quantity Unit. Code List Agency Name. Text**: The name of the agency which maintains the quantity unit code list.

```xml
<cbc:PackQuantity unitCode="EA">1</cbc:PackQuantity>
```
Text (Name)

Text. Content

A character string (i.e. a finite set of characters) generally in the form of words of a language.

Language. Identifier

The identifier of the language used in the corresponding text string.

Language. Locale. Identifier

The identification of the locale of the language.

Note: The Representation term “Text” is not part of the tag name (UBL naming rule)

<cbc:Description>Nokia Mobile telephone – Type ABC</cbc:Description>
UBL Design

Codes and Identifiers
Sets of Codes

- A special set of possible values.
- The code value refers to something else.
- May be symbolic representations:
  - Initialisms (“EU”),
  - Acronyms (“SARS”),
  - Apocopations (“Reefer”) or
  - Arbitrary abbreviations (“XML”).
- But maybe not...
  - Both ISO code “CNY” and “156” identify the Chinese Yuan Renminbi.
# Codes and Values

## Units of Measurement codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>Capsule</td>
</tr>
<tr>
<td>BA</td>
<td>Barrel</td>
</tr>
<tr>
<td>BB</td>
<td>Bobbin</td>
</tr>
<tr>
<td>BC</td>
<td>Bottlecrate / bottlerack</td>
</tr>
<tr>
<td>BF</td>
<td>Balloon, non-protected</td>
</tr>
<tr>
<td>BG</td>
<td>Bag</td>
</tr>
<tr>
<td>BH</td>
<td>Bunch</td>
</tr>
<tr>
<td>BI</td>
<td>Bin</td>
</tr>
<tr>
<td>BJ</td>
<td>Bucket</td>
</tr>
<tr>
<td>BK</td>
<td>Basket</td>
</tr>
</tbody>
</table>

## Types of packaging codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMT</td>
<td>kilometre</td>
</tr>
<tr>
<td>C45</td>
<td>nanometre</td>
</tr>
<tr>
<td>C52</td>
<td>picometre</td>
</tr>
<tr>
<td>A71</td>
<td>femtometre</td>
</tr>
<tr>
<td>A45</td>
<td>decametre</td>
</tr>
<tr>
<td>NMI</td>
<td>nautical mile</td>
</tr>
<tr>
<td>A11</td>
<td>foot</td>
</tr>
<tr>
<td>A12</td>
<td>astronomical unit</td>
</tr>
<tr>
<td>C63</td>
<td>parsec</td>
</tr>
<tr>
<td>AK</td>
<td>fathom</td>
</tr>
<tr>
<td>X1</td>
<td>chain</td>
</tr>
<tr>
<td>INH</td>
<td>inch</td>
</tr>
<tr>
<td>M7</td>
<td>micro-inch</td>
</tr>
<tr>
<td>FOT</td>
<td>foot</td>
</tr>
<tr>
<td>YRD</td>
<td>yard</td>
</tr>
</tbody>
</table>
Identifiers

- Have meaning in their own right.
  - Don’t have to reference anything.
- May be a constrained set of values:
  - Set of Driver’s License numbers issued.
- May be symbolic representations (like codes).
- Or (effectively) not…
  - The values for identifying a phone number.
- We use codes as identifiers:
  - Digits in phone numbers are codes for regions.
- But not identifiers as codes:
  - What does your Driver’s License number refer to?
**What are these?**

<table>
<thead>
<tr>
<th>ABCU123456XYZ</th>
<th>Waybill Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCU123456789000</td>
<td></td>
</tr>
<tr>
<td>ABCU123456789001</td>
<td></td>
</tr>
</tbody>
</table>

**And these?**

<table>
<thead>
<tr>
<th>Shipping Container Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCLU1234567</td>
</tr>
<tr>
<td>NHKL7654321</td>
</tr>
<tr>
<td>HPGL8877665</td>
</tr>
</tbody>
</table>
UBL Specification
Specifying Documents

• We need a range of metadata to support the explanation of meanings.
  - Names are not enough.

• We define these in models.
UBL Specification
Models and Modeling
Component Models

• A generalized, conceptual model.
  - “webs” or networks.
  - Domain Model, Enterprise Data Model.

• Not a model of one document… but,

• A model of Information Entities usable in all documents.
  - and databases, etc.

• Complexity depends on the number of the business rules.
A Component Model
UBL’s Common Components
UBL Specification
Assembling Components into Documents
Documents are Hierarchical

- Inverse tree structure.
- Can be represented in a 2D space.
  - That’s why markup languages need just `<start>` and `</end>` tags.
- Nesting imposes meaning.
  - When “Contact” contains “Name” we know that the name is that of the contact.
  - Parent (genealogy) gives contact of use.
Hierarchy not Network
A Document Assembly Model
## Another Document Assembly Model

<table>
<thead>
<tr>
<th>UBL Name</th>
<th>Dictionary Entry Name</th>
<th>Cardinality</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerAssignedAccountId</td>
<td>Customer Party. Customer Assigned_AccAccount Identifier.</td>
<td>0..1</td>
<td>An identifier for the Customer's account, assigned by the Customer itself.</td>
</tr>
<tr>
<td>SupplierAssignedAccountId</td>
<td>Customer Party. Supplier Assigned_AccAccount Identifier.</td>
<td>0..1</td>
<td>An identifier for the Customer's account, assigned by the Supplier.</td>
</tr>
<tr>
<td>AdditionalAccountID</td>
<td>Customer Party. Additional_AccAccount Identifier.</td>
<td>0..n</td>
<td>An identifier for the Customer's account, assigned by a third party.</td>
</tr>
<tr>
<td>Party</td>
<td>Customer Party. Party</td>
<td>0..1</td>
<td>An association to Party.</td>
</tr>
<tr>
<td>DeliveryContact</td>
<td>Customer Party. Delivery_AccContact. Contact</td>
<td>0..1</td>
<td>An association to Delivery Contact.</td>
</tr>
<tr>
<td>AccountingContact</td>
<td>Customer Party. Accounting_AccContact. Contact</td>
<td>0..1</td>
<td>An association to Accounting Contact (Customer).</td>
</tr>
<tr>
<td>BuyerContact</td>
<td>Customer Party. Buyer_AccContact. Contact</td>
<td>0..1</td>
<td>An association to Buyer Contact.</td>
</tr>
</tbody>
</table>
Creating a Document Assembly

- Each type of document usually requires its own document assembly model.
- We create a document assembly model by:
  1. Choosing an “entry point” into the Document Component Model.
     - The entry point is a structure that will form the root of the document tree.
  2. Following a pathway through the component model
     - guided by business rules for the document required.
Assembling Associations

• If an association is mandatory:
  - must follow the path
  - the associated ABIE must be in the document.

• If an association is optional:
  - may follow the path
  - the associated ABIE can be in the document if required by business rules.

• Cardinality of associations controls the depth of the document hierarchy.
UBL Specification
Assembly Patterns
Patterns for Document Assembly

• Document assembly models express business rules based on the context of use.

• Many assemblies share common structures.
  - There are patterns of assemblies.
  - a Book, a Contract, an Order.

• Good designers employ these patterns...
  - Because processes will find their documents familiar.
The “Book” Assembly Pattern

- Foreword
- Preface
- Introduction
- Chapter
- Appendix
- Bibliography
- Index
- Text Book
- Section
- Exercise
Reusing Assembly Patterns

- Assembly patterns encourages the use of common libraries of components.
  - Reusing patterns from elsewhere in the model
  - Adopting patterns from outside
- The challenge is customizing these patterns to suit specific requirements.
COMPOSING MEANINGFUL DOCUMENTS WITH UBL

Tim McGrath
Open Standards 2008: Composability within SOA Symposium
Date: 28 April – 1 May
Location: Santa Clara Marriott
COMPOSING MEANINGFUL DOCUMENTS
WITH UBL

- an approach to design and customization of UBL to satisfy specific requirements

UBL is establishing itself as the language of choice for a variety of electronic commerce applications. It is particularly well suited for use in business web services and service oriented architectures. The use of UBL for such environments ensures that services remain meaningful and worthy in a variety of different compositions. Inevitably, these applications may require some refinement of the standard structures.

This presentation explains some of the design principles behind UBL and how both the standard components and the principles behind them can be adapted to suit the requirements of different business processes. We will explore a case study involving the Danish Government's OIOUBL and Service Oriented Infrastructure project.

Composing different services into a meaningful business process requires interoperability of the information contained in the documents exchanged. This tutorial will discuss the opportunities for utilizing the OASIS Universal Business Language (UBL) as the glue to bind these services together at a business data level.

This tutorial explains the design principles behind UBL and how it can be adapted to suit the requirements of different business processes. These composite applications will require some customization of the standard structures and the creation of new document types.

The objective is to enable participants to understand the issues involved in customizing UBL and how to develop specific implementations that suit the requirements of their context of use. The target audience is system analysts and developers responsible for the implementation of document interfaces.
Outline

- Background to Document Design
- UBL Design
- UBL Specification
- Case Study Scenario
- Designing Customizations
- Specifying Customizations
- Validating Customizations
- Summary and Feedback
Introductions
Who are we and why are we here?
Background to Document Design
What are Documents?

• For nearly three thousand years information has been organized in purposeful and self-contained packages.
  - We call them documents.
• Using documents for exchanging business information is an old idea.
  - The technology for encoding and exchanging documents has profoundly changed
  but
  - The concept of a document has remained surprisingly stable.
Using documents for exchanging business information is natural and intuitive.

Administrative documents
Early Dynastic III, about 2500 BC
Probably from Sharuppak (Fara)
Left: the governor receives 14 shekels of silver from 5 persons, and 40 shekels remain unpaid by 13 other persons. The total amount of silver is 1 mina.
How Documents Work

• Documents are self-contained collections of information.
  - Interfaces for people.
  - Interfaces to business processes.
• To be meaningful documents must be defined using a common language.
  - Not for information content but
  - For describing the meaning (semantics) of the information content.
• The Universal Business Language is a semantic document description language.
A document exchange consists of both processes and the documents (information) they produce and consume.

By understanding the processes
- we learn what kinds of information is needed.

By understanding the document
- we learn what kinds of processes are possible.
Interoperability

• A basic requirement for two businesses to conduct business is that their business systems interoperate.
• Easy to express but hard to achieve.
• Variations in strategies, technology platforms, legacy applications, business processes, and terminology.
• Interoperability doesn’t require that two systems be identical.
  – Just they have common understanding of the meanings of things (semantics)
UBL Design
UBL Design

ebXML Core Components
• ebXML Core Components Technical Specification:
  - ISO 15000-5.
  - UN/CEFACT
• How to describe core components of business documents in a syntax independent manner.
• UBL is the first set of integrated XML document specifications based on CCTS.
  - UN/CEFACT recognizes UBL as appropriate first-generation XML documents for eBusiness.
CCTS Principles

Core

- Aggregate Core Component (BCC)
- Basic Core Component (BCC)
  - is of type
  - is defined in context as

Business

- Aggregate Business Information Entity (ABIE)
- Basic Business Information Entity (BBIE)
  - is defined in context as

These are implied concepts

UBL publishes these

Core Component Type (CCT)

Core Business Message/Document

These are implied concepts
**Business Information Entities**

- Reusable building blocks for the exchange of information ("Core Components") in a specific Business Context.
- **Aggregate BIE**
  - A collection of related pieces of business information that together convey a distinct business meaning.
- **Association BIE**
  - Is associated to an Aggregate Business Information Entity, which describes its structure.
- **Basic BIE**
  - A singular business characteristic of a specific aggregation.
Basic Business Information Entities

- A singular business characteristic in a specific business context.
- Individual pieces of information.

```
-ABIE
An Address
  # «BBIE» Street Name: Name [0..1]
  # «BBIE» City Name: Name [0..1]
```
Aggregate Business Information Entities

- A collection of related pieces of business information.
- A container of BBIEs and ASBIEs to other ABIEs.
Association
Business Information Entities

• Represents an ABIE in a specific business context.
• An ABIE contained within another ABIE.
UBL Design
Naming Information Entities
Names

• Meaningful Information Entity names.
  - promote a common understanding.
  - encourage reuse.
• May need different names in different modeling artifacts.
  - XML Tag may not be the modeling or the presentation name.
Rules for Consistent Naming

- Names should be grammatically consistent.
  - Singular not plurals
- Names should not be ambiguous.
  - Is "Supplier’s ID" the same as "ID of Supplier"?
- Check names are not semantically misleading.
  - "Product Number" may not be a number, so "Product Identifier" is better.
- Qualify names to clarify the context of use.
  - "Customer Party" is more meaningful than "Customer".
ebXML Naming Rules

- ebXML Core Components naming rules are an application of ISO 11179.
- All Information Entities get a Dictionary Entry Name (DEN).
- Dictionary Entry Names are globally unique.
- They are the ‘key’ or unique identifier for any Information Entity.
Dictionary Entry Names

{qualifier} Object Class Term. ". ",
{qualifier} Property Term. ". ",
{qualifier} Representation Term

• Object Class Term
  - The object class to which it belongs.
• Property Term
  - Reflecting its function as a property or distinguishing characteristic of the object class.
• Representation Term
  - How it is represented.
UBL Design

Representation Terms
<table>
<thead>
<tr>
<th>Content</th>
<th>A number of monetary units specified in a currency where the unit of currency is explicit or implied.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount. Currency Identifier</td>
<td>The currency of the amount.</td>
</tr>
<tr>
<td>Amount. Currency Code List Version Identifier</td>
<td>The version of the UN/ECE rec. 9 code list.</td>
</tr>
</tbody>
</table>

```xml
<cbc:TaxAmount currencyID="DKK">1262.50</cbc:TaxAmount>
```
## Binary Object

(Binary, Picture, Sound, Video)

<table>
<thead>
<tr>
<th>Binary Object</th>
<th>Content</th>
<th>A set of finite-length sequences of binary octets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Object</td>
<td>Format</td>
<td>The format of the binary content.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Object</td>
<td>Mime Code</td>
<td>The mime type of the binary object.</td>
</tr>
<tr>
<td>Encoding</td>
<td></td>
<td>Specifies the decoding algorithm of the binary</td>
</tr>
<tr>
<td>Code</td>
<td></td>
<td>object.</td>
</tr>
<tr>
<td>Binary Object</td>
<td>Character</td>
<td>The character set of the binary object if the mime</td>
</tr>
<tr>
<td>Set Code</td>
<td></td>
<td>Type is text.</td>
</tr>
<tr>
<td>Binary Object</td>
<td>Uniform</td>
<td>The Uniform Resource Identifier that identifies</td>
</tr>
<tr>
<td>Resource</td>
<td></td>
<td>where the binary object is located.</td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Object</td>
<td>Filename</td>
<td>The filename of the binary object.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```xml
<cbc:EmbeddedDocumentBinaryObject
    mimeCode="application/pdf"
    encodingCode="Base64"
    characterSetCode="UTF-8">
    brIG0Ng53KHKazOQNQYmpFHk9Ukkg2jPB9fJ1DBUasdgdartujjoprr
    wer34578deewcowetytuytyuy64342THyB4K5SYMBZcpMkshU0UzjL
    HSDEWFdrefsdsl//
</cbc:EmbeddedDocumentBinaryObject>
```
<table>
<thead>
<tr>
<th>Code. Content</th>
<th>A character string (letters, figures or symbols) that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code List Identifier</td>
<td>The identification of a list of codes.</td>
</tr>
<tr>
<td>Code List Agency Identifier</td>
<td>An agency that maintains one or more code lists.</td>
</tr>
<tr>
<td>Code List Agency Name Text</td>
<td>The name of the agency that maintains the code list.</td>
</tr>
</tbody>
</table>

```xml
<cbc:ItemClassificationCode listID="UNSPSC" listVersionID="7.0401">43191501</cbc:ItemClassificationCode>
```
<table>
<thead>
<tr>
<th>Code List Name Text</th>
<th>The name of a list of codes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code List Version Identifier</td>
<td>The version of the code list.</td>
</tr>
<tr>
<td>Code Name Text</td>
<td>The textual equivalent of the code content.</td>
</tr>
<tr>
<td>Language Identifier</td>
<td>The identifier of the language used in the corresponding text string.</td>
</tr>
<tr>
<td>Code List Uniform Resource Identifier</td>
<td>The Uniform Resource Identifier that identifies where the code list is located.</td>
</tr>
<tr>
<td>Code List Scheme Uniform Resource Identifier</td>
<td>The Uniform Resource identifier that identifies where the code list scheme is located.</td>
</tr>
</tbody>
</table>

```xml
<cbc:ItemClassificationCode listID="UNSPSC" listVersionID="7.0401">
  #43191501
</cbc:ItemClassificationCode>
```
Date Time
(Date, Time)

Date Time, Content
The particular point in the progression of time.

Date Time, Format,
Text
The format of the date/time content.

<cbc:StartDate>2007-01-25</cbc:StartDate>
### Identifier

A character string used to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects within the same scheme.

<table>
<thead>
<tr>
<th>Identification Scheme</th>
<th>Name, Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier</td>
<td>Content</td>
</tr>
</tbody>
</table>

```xml
<cbc:ID schemeID="GTIN">5712345780121</cbc:ID>
```
## Identifier (contd.)

<table>
<thead>
<tr>
<th>Identifier Scheme</th>
<th>The identification of the agency that maintains the identification scheme.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agency Identifier</td>
<td>The name of the agency that maintains the identification scheme.</td>
</tr>
<tr>
<td>Identification Scheme</td>
<td>The version of the identification scheme.</td>
</tr>
<tr>
<td>Data Uniform Resource Identifier</td>
<td>The Uniform Resource Identifier that identifies where the identification scheme data is located.</td>
</tr>
<tr>
<td>Uniform Resource Identifier</td>
<td>The Uniform Resource Identifier that identifies where the identification scheme is located.</td>
</tr>
</tbody>
</table>
## Indicator

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicator</td>
<td>Format</td>
<td>Whether the indicator is numeric,</td>
</tr>
<tr>
<td></td>
<td>Text</td>
<td>textual or binary.</td>
</tr>
</tbody>
</table>

```xml
<cbc:ChargeIndicator>false</cbc:ChargeIndicator>
```
## Measure

<table>
<thead>
<tr>
<th>Measure Content</th>
<th>The numeric value determined by measuring an object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure Unit Code</td>
<td>The type of unit of measure.</td>
</tr>
<tr>
<td>Measure Unit Code List Version</td>
<td>The version of the measure unit code list.</td>
</tr>
<tr>
<td>Identifier</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<cbc:GrossWeightMeasure unitCode="KGS">21408.79</cbc:GrossWeightMeasure>
```
Numeric
(Value, Rate, Percent)

<table>
<thead>
<tr>
<th>Numeric. Content</th>
<th>Numeric information that is assigned or is determined by calculation, counting or sequencing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric Format</td>
<td>Whether the number is an integer, decimal, real number or percentage.</td>
</tr>
<tr>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<cbc:PackSizeNumeric> 1 </cbc:PackSizeNumeric>
```
### Quantity

<table>
<thead>
<tr>
<th>Quantity Unit Code</th>
<th>A counted number of non-monetary units possibly including fractions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Agency Name</td>
<td>The unit of the quantity.</td>
</tr>
<tr>
<td>List Identifier</td>
<td>The quantity unit code list.</td>
</tr>
<tr>
<td>Identifier</td>
<td>The identification of the agency that maintains the quantity unit code list.</td>
</tr>
<tr>
<td>Text</td>
<td>The name of the agency which maintains the quantity unit code list.</td>
</tr>
</tbody>
</table>

```xml
<cbc:PackQuantity unitCode="EA">1</cbc:PackQuantity>
```
The identification of the locale of the language.

Language. Locale Identifier
The identification of the locale of the language.

Language. Identifier
The identifier of the language used in the corresponding text string.

Text. Content
A character string (i.e. a finite set of characters) generally in the form of words of a language.

Note: The Representation term “Text” is not part of the tag name (UBL naming rule)

<n:cbc:Description>Nokia Mobile telephone – Type ABC</n:cbc:Description>
UBL Design
Codes and Identifiers
Sets of Codes

- A special set of possible values.
- The code value refers to something else.
- May be symbolic representations:
  - Initialisms ("EU"),
  - Acronyms ("SARS"),
  - Apocopations ("Reefer") or
  - Arbitrary abbreviations ("XML").
- But maybe not...
  - Both ISO code "CNY" and "156" identify the Chinese Yuan Renminbi.
### Codes and Values

#### Types of packaging codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Capsule</td>
<td>KMT (kilometre)</td>
</tr>
<tr>
<td>BA</td>
<td>Balloon</td>
<td>C45 (centimetre)</td>
</tr>
<tr>
<td>BB</td>
<td>Bollard</td>
<td>C52 (micrometre)</td>
</tr>
<tr>
<td>BC</td>
<td>Bottleneck / bollard</td>
<td>D41 (nanometre)</td>
</tr>
<tr>
<td>BF</td>
<td>Bag</td>
<td>A71 (femtometre)</td>
</tr>
<tr>
<td>BH</td>
<td>Bunch</td>
<td>A45 (decimetre)</td>
</tr>
<tr>
<td>BJ</td>
<td>Bin</td>
<td>NMI (nautical mile)</td>
</tr>
<tr>
<td>BK</td>
<td>Basket</td>
<td>A11 (astronomical unit)</td>
</tr>
<tr>
<td>BS</td>
<td>Bissel</td>
<td>A12 (millimetre)</td>
</tr>
<tr>
<td>X1</td>
<td>Chain</td>
<td>D1 (decimetre)</td>
</tr>
<tr>
<td>X2</td>
<td>Chain</td>
<td>A4 (centimetre)</td>
</tr>
<tr>
<td>TRM</td>
<td>Ton</td>
<td>X1 (chain)</td>
</tr>
<tr>
<td>MT</td>
<td>Micro-ton</td>
<td>TRM (ton)</td>
</tr>
<tr>
<td>FOT</td>
<td>Foot</td>
<td>MT (micro-ton)</td>
</tr>
<tr>
<td>YRD</td>
<td>Yard</td>
<td>FOT (foot)</td>
</tr>
</tbody>
</table>

#### Units of Measurement codes

- KMT: kilometre
- C45: centimetre
- C52: micrometre
- D41: nanometre
- A71: femtometre
- A45: decimetre
- NMI: nautical mile
- A11: astronomical unit
- A12: millimetre
- D1: decimetre
- A4: centimetre
- TRM: ton
- MT: micro-ton
- FOT: foot
- YRD: yard
Identifiers

- Have meaning in their own right.
  - Don't have to reference anything.
- May be a constrained set of values:
  - Set of Driver's License numbers issued.
- May be symbolic representations (like codes).
- Or (effectively) not…
  - The values for identifying a phone number.
- We use codes as identifiers:
  - Digits in phone numbers are codes for regions.
- But not identifiers as codes:
  - What does your Driver's License number refer to?
What are these?

Waybill Numbers

ABCUIJKL123456XYZ
ABCU123456789000
ABCU123456789001

And these?

Shipping Container Numbers

OCLU1234567
NHKL7654321
HPGL8877665
UBL Specification
Specifying Documents

- We need a range of metadata to support the explanation of meanings.
  - Names are not enough.
- We define these in models.
UBL Specification
Models and Modeling
A generalized, conceptual model.
- "webs" or networks.
- Domain Model, Enterprise Data Model.

Not a model of one document…
but,

A model of Information Entities usable in all documents.
- and databases, etc.

Complexity depends on the number of the business rules.
A Component Model

- Common Library::Item
- Shipment
- Goods Item
- Consignment
- Transport Equipment
- Transport Means

- Procurement Library::Line Item
UBL Specification
Assembling Components into Documents
Documents are Hierarchical

- Inverse tree structure.
- Can be represented in a 2D space.
  - That's why markup languages need just `<start>` and `<end>` tags.
- Nesting imposes meaning.
  - When “Contact” contains “Name” we know that the name is that of the contact.
  - Parent (genealogy) gives contact of use.
A Document Assembly Model

«ABIE» Common Library: Party
- «BBIE» Mark Care Indicator: Indicator [0..1]
- «BBIE» Mark Attention Indicator: Indicator [0..1]
- «BBIE» Website URI: Identifier [0..1]
- «BBIE» Logo Reference ID: Identifier [0..1]
- «BBIE» End Point ID: Identifier [0..1]

«ABIE» Common Library: Contact
- «BBIE» ID: Identifier [0..1]
- «BBIE» Name: Name [0..1]
- «BBIE» Telephone: Text [0..1]
- «BBIE» Telefax: Text [0..1]
- «BBIE» Electronic Mail: Text [0..1]
- «BBIE» Note: Text [0..1]
## Another Document Assembly Model

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Party</td>
<td>An identifier for the Customer's account, assigned by the Customer's identifier.</td>
</tr>
<tr>
<td>Supplier Party</td>
<td>An identifier for the Customer's account, assigned by the Supplier's identifier.</td>
</tr>
<tr>
<td>Additional Account ID</td>
<td>An identifier for the Customer's account, assigned by a third party.</td>
</tr>
<tr>
<td>Party</td>
<td>An association to Party.</td>
</tr>
<tr>
<td>Delivery Contact</td>
<td>An association to Delivery Contact.</td>
</tr>
<tr>
<td>Accounting Contact</td>
<td>An association to Accounting Contact (Customer).</td>
</tr>
<tr>
<td>Buyer Contact</td>
<td>An association to Buyer Contact.</td>
</tr>
</tbody>
</table>
Creating a Document Assembly

- Each type of document usually requires its own document assembly model.
- We create a document assembly model by:
  1. Choosing an “entry point” into the Document Component Model.
     - The entry point is a structure that will form the root of the document tree.
  2. Following a pathway through the component model
     - guided by business rules for the document required.
- If an association is mandatory:
  - must follow the path
  - the associated ABIE must be in the document.
- If an association is optional:
  - may follow the path
  - the associated ABIE can be in the document if required by business rules.
- Cardinality of associations controls the depth of the document hierarchy.
UBL Specification
Assembly Patterns
Patterns for Document Assembly

- Document assembly models express business rules based on the context of use.
- Many assemblies share common structures.
  - There are patterns of assemblies.
  - a Book, a Contract, an Order.
- Good designers employ these patterns...
  - Because processes will find their documents familiar.
The “Book” Assembly Pattern

Diagram:

- Text Book
- Foreword
- Preface
- Introduction
- Chapter
- Appendix
- Bibliography
- Index
- Section
- Exercise

Relationships:

- 0..1 Text Book
- 0..1 Foreword
- 0..1 Preface
- 0..1 Introduction
- 0..1 Chapter
- 0..1 Appendix
- 0..1 Bibliography
- 0..1 Index
- 0..1 Section
- 0..1 Exercise
Reusing Assembly Patterns

- Assembly patterns encourages the use of common libraries of components.
  - Reusing patterns from elsewhere in the model
  - Adopting patterns from outside
- The challenge is customizing these patterns to suit specific requirements.