www.oasis-open.org/events/symposium_2005

The Future of XML Vocabularies

SYMPOSIUM

24 April: Tutorials
8:30 AM – 12:00 PM
New Orleans Marriott

Creating UBL Conformant Schema Tutorial

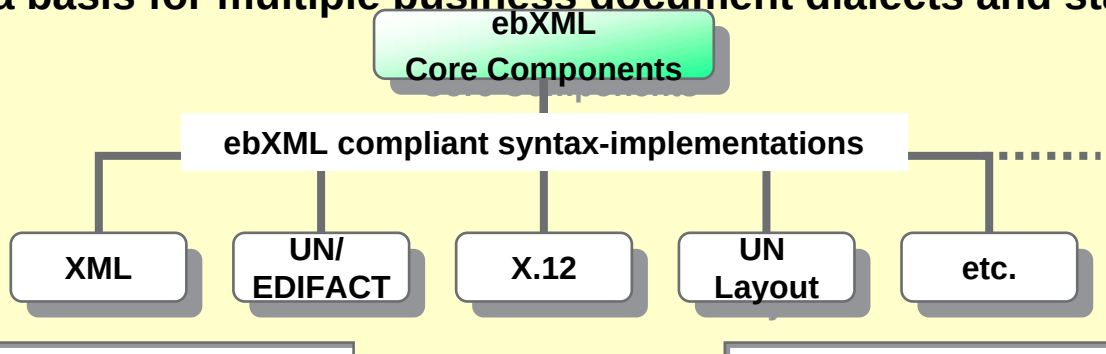
Mark Crawford
Senior Research Fellow
LMI Government Consulting
mcrawford@lmi.org

Agenda

- UBL Overview
- The Modeling Methodology
- Core Components and Business Information Entities
- XML NDR
- Creating the Schema
- Customizing UBL

Why UBL?

ebXML Core Components are „syntax neutral“,
it will be a basis for multiple business document dialects and standards

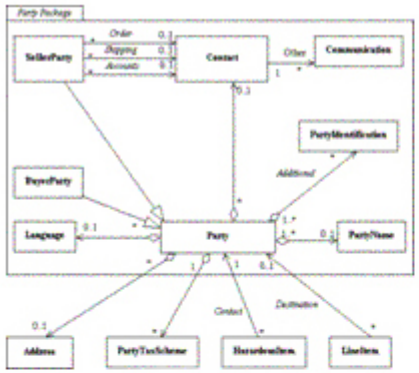


- **But we must have concrete standard XML syntax to enable wide use and cheap commercial software**
- **Given a concrete XML syntax for business, users will adopt it**

UBL is developing XML business document design rules, XML syntax based on ebXML core component (CC) structures and ebXML (UN/CEFACT) CC compliant XML document schemas

©Mark Crawford, LMI Government Consulting, April 2005

The UBL Development Approach



component model

Manual assembly of document structures as BIEs

spreadsheet assembly model

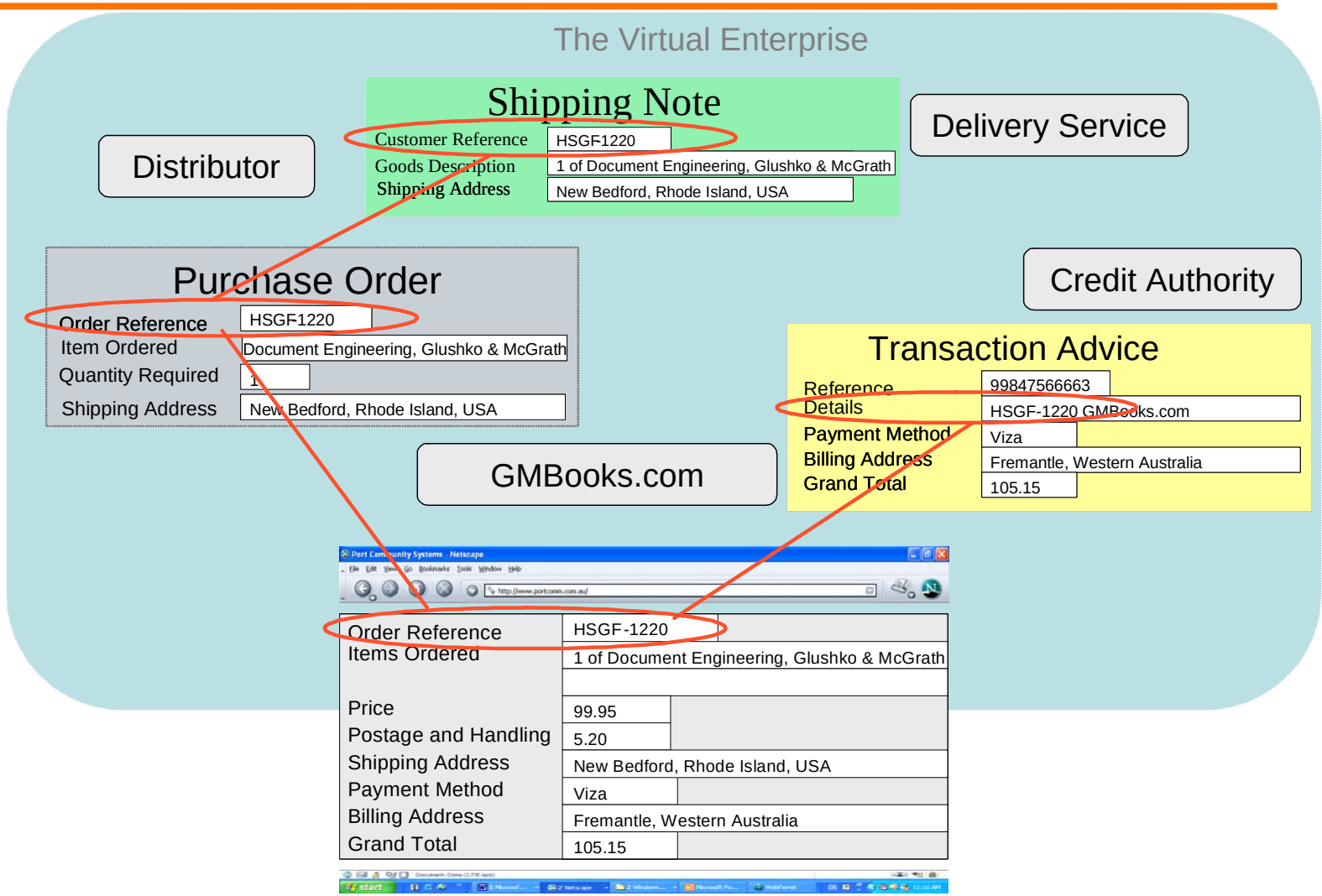
Automated transformation according to UBL Naming and Design Rules

schemas

Automated transformation

implementation model

The Interoperability Challenge



Customer's View of Buying a Book

Taken from: Document Engineering, Glushko and McGrath, MIT Press, 2005©

Understanding Documents

LMI

- Interoperability means understanding the meaning of documents and their information components.
- This is facilitated when their **semantics**, **structure** and **syntax** conform to standard patterns.
- XML has become the preferred **syntax** pattern for representing information in documents.
- Now we need to define common patterns for the **semantics** and **structure** of business documents.

©Mark Crawford, LMI Government Consulting, April 2005

Document Engineering

LMI

- A new discipline needed for analyzing and designing new business documents.
- Synthesizes complementary ideas from business analysis, task analysis, document analysis and data analysis.
- The OASIS UBL TC has document engineered re-usable semantic and structural patterns for common business requirements…
- … to create a **Universal Business Language**.

Creating Conceptual Component Models

LMI

Business Operations View

UML and spreadsheets

Conceptual models showing all possible associations

Analysis

Functional Service view

Schemas

XML Schema Libraries

Database Schemas

EDI Message Definitions

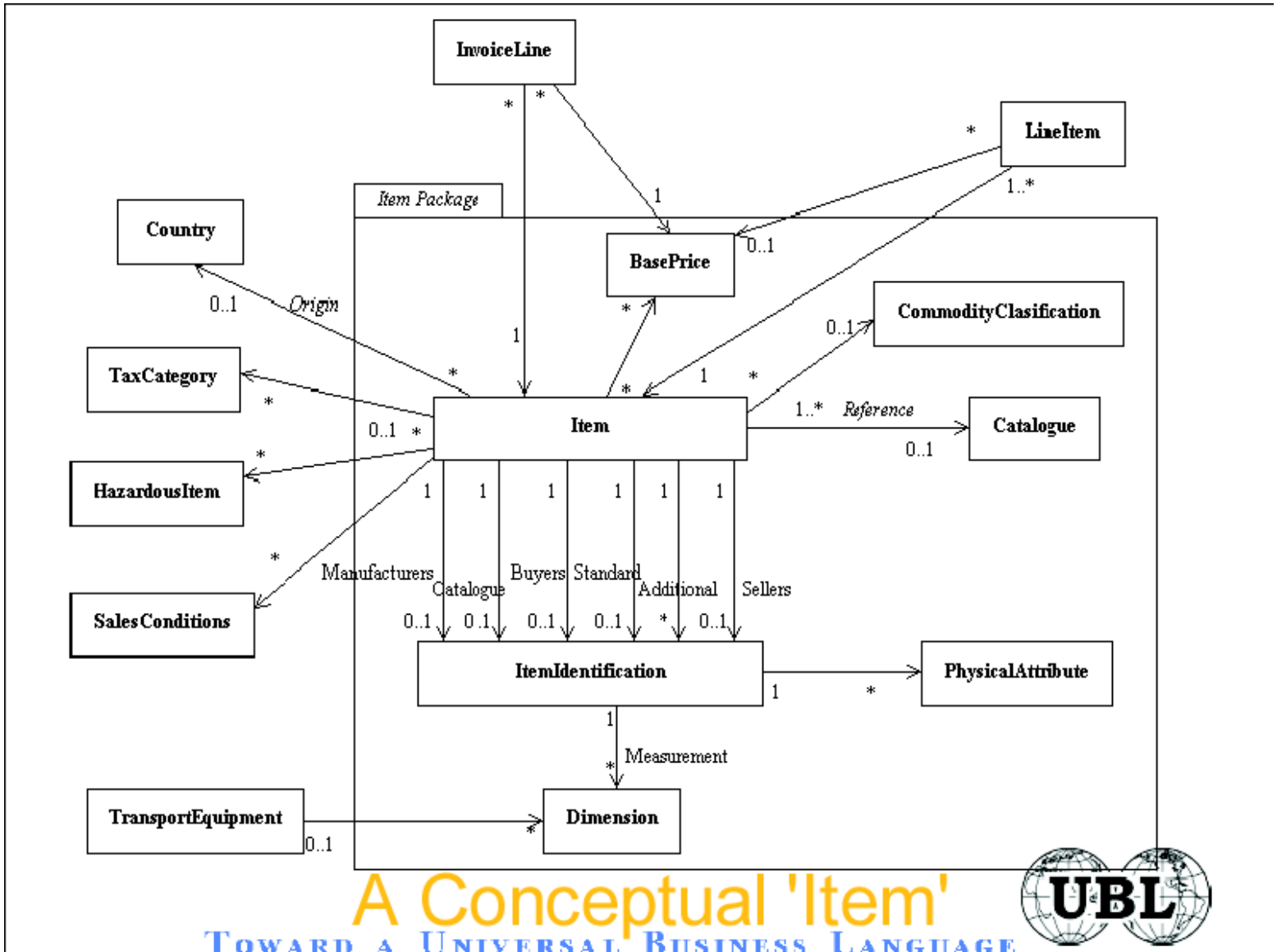
Business Document Models

The Real World

Messages/Documents

Limited interoperability

Component Model for "Item"



A Conceptual 'Item'

TOWARD A UNIVERSAL BUSINESS LANGUAGE

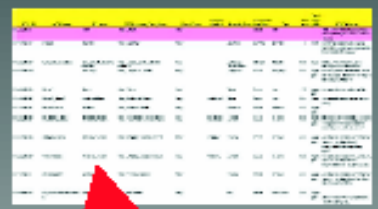
Creating the Document Structures

LMI

Business Operations View

UML and spreadsheets

Design



Analysis

XML Schema Libraries

Database Schemas

EDI Message Definitions

Business Document Models

Functional Service view

Schemas

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

Limited interoperability

A Spreadsheet Sample - Item

LMI

UBL Name	BIE Dictionary Entry Name	Occurrence	BIE Type	UBL Definition
Item	Item. Details		ABIE	information directly relating to an item
Description	Item. Description. Text	0..1	BBIE	a free form field that can be used to give a text
PackQuantity	Item. Pack. Quantity	0..1	BBIE	the unit packaging quantity.
PackSizeQuantity	Item. Pack_Size. Quantity	0..1	BBIE	the number of items in a pack.
FromCatalogueIndicator	Item. From Catalogue. Indicator	0..1	BBIE	an indicator that denotes whether or not the item was
BuyersItemIdentification	Item. Buyers_ Item Identification	0..1	ASBIE	associates the item with its identification according to the buyers system.
SellersItemIdentification	Item. Sellers_ Item Identification	0..1	ASBIE	associates the item with its identification according to the sellers system.
ManufacturersItemIdentification	Item. Manufacturers_ Item Identification	0..1	ASBIE	associates the item with its identification according to the manufacturers system.
StandardItemIdentification	Item. Standard_ Item Identification	0..1	ASBIE	associates the item with its identification according to a standard system.
CatalogueItemIdentification	Item. Catalogue_ Item Identification	0..1	ASBIE	associates the item with its identification according to a cataloging system.
AdditionalItemIdentification	Item. Additional_ Item Identification	0..n	ASBIE	associates the item with other identification means
CatalogueReference	Item. Catalogue_ Reference	0..1	ASBIE	associates the item with the catalogue from which the item was selected.
OriginCountry	Item. Origin_ Country	0..1	ASBIE	associates the item with its country of origin
CommodityClassification	Item. Commodity Classification	0..1	ASBIE	associates the item with its classification(s) according to a commodity classifying system.
SalesConditions	Item. Sales Conditions	0..n	ASBIE	associates the item with sales conditions appertaining to it.
HazardousItem	Item. Hazardous Item	0..n	ASBIE	associates the item with its hazardous item information.
TaxCategory	Item. Tax Category	0..n	ASBIE	associates the item with one or more taxes
BasePrice	Item. Base Price	0..n	ASBIE	associates the item with one or more base prices.

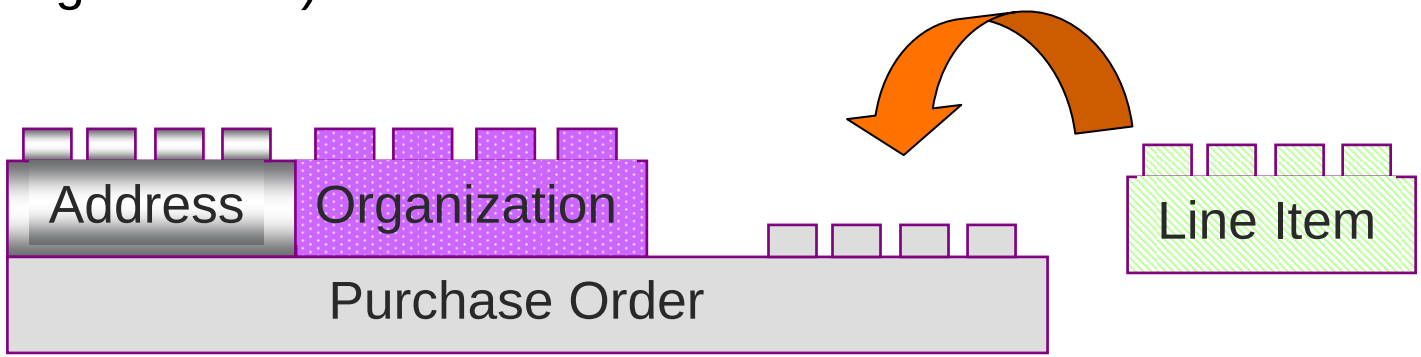
The Core Components Technical Specification

LMI

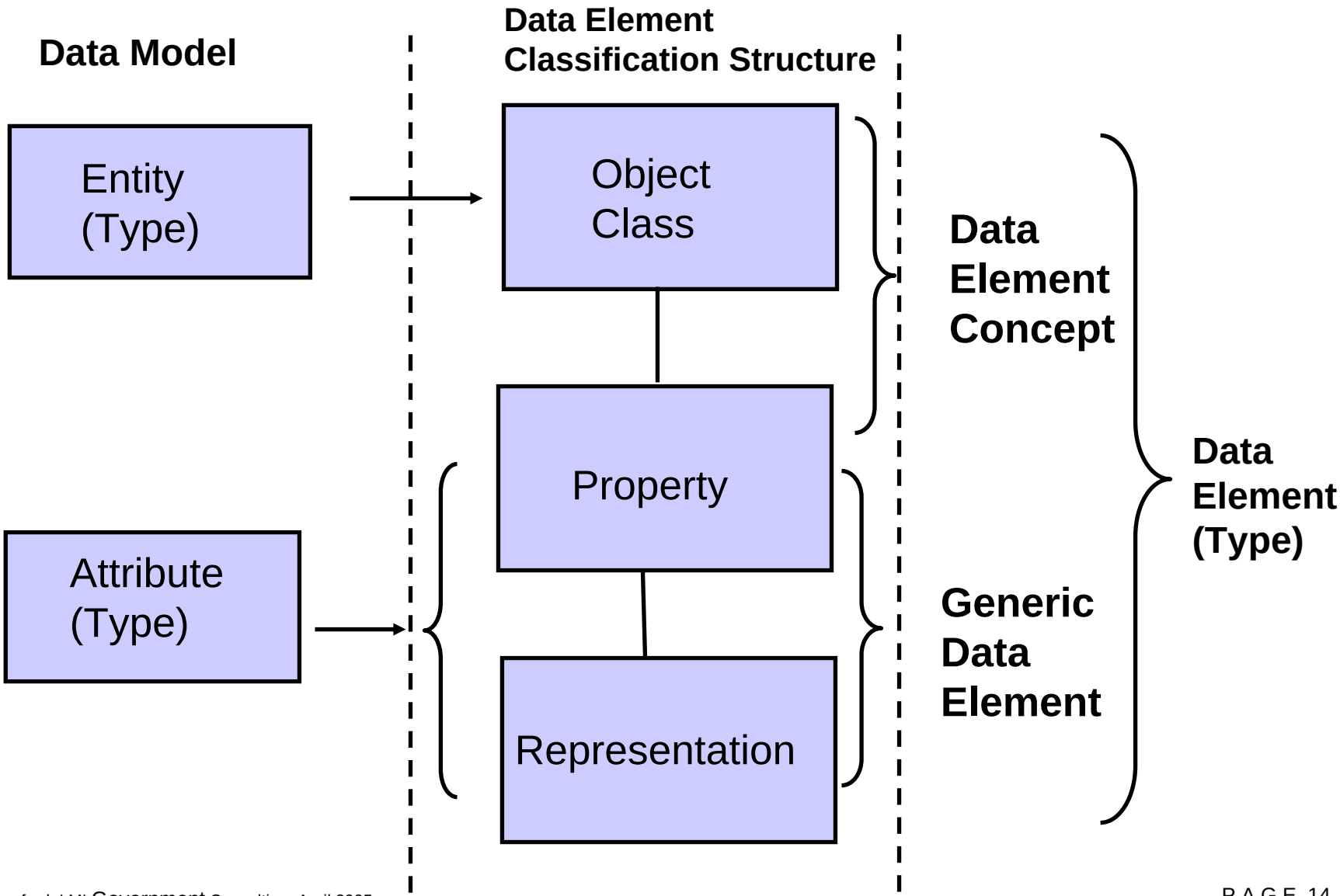
ISO/TS 15000-5:2004
electronic business Extensible Markup Language (ebXML) -- Part 5: Core Components Technical Specification (ebCCTS)

ISO 15000-5 The Core Components Technical Specification (CCTS)

- Implementation rules for ISO 11179 parts 4 and 5
- A methodology for developing a common set of semantic building blocks representing general types of business data
  - Adds structure and consistency to database constructs
  - Provides for the creation of new business vocabularies and restructuring of existing business vocabularies
  - Is flexible and interoperable
- Defines a **syntax-neutral** meta-model for business semantics (meaning of words)



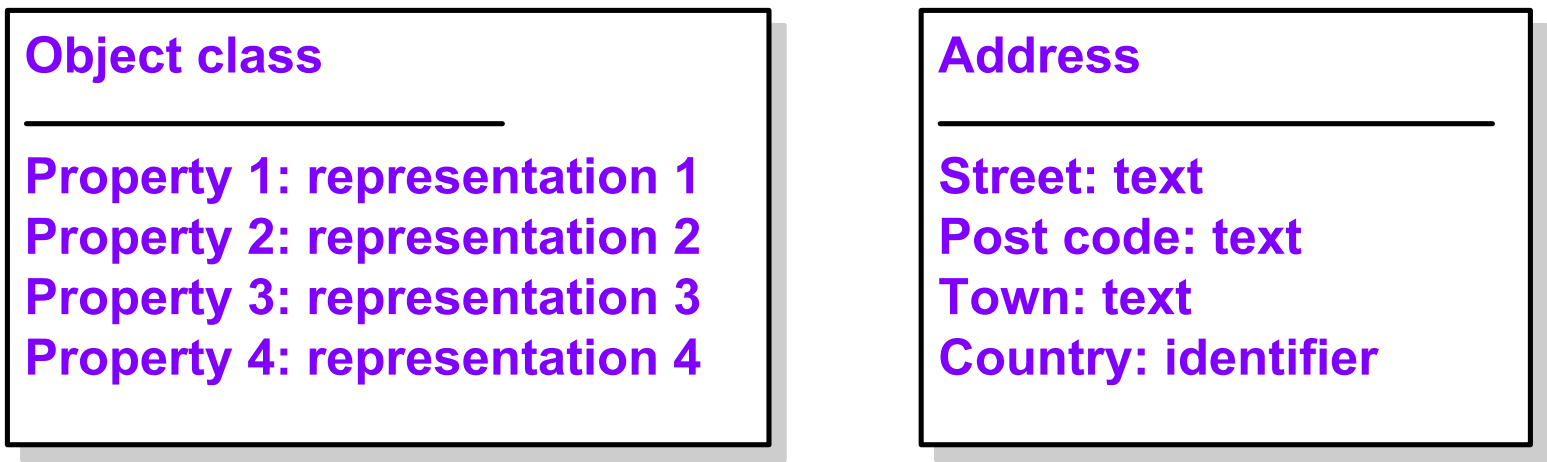
ISO 11179 Data Constructs



The Baseline – ISO 15000-5 Follows ISO 11179

LMI

- This is basic object-oriented “good stuff”



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

©Sun Microsystems, Eve Mailer, Dec 2002
©Mark Crawford, LMI Government Consulting, April 2005

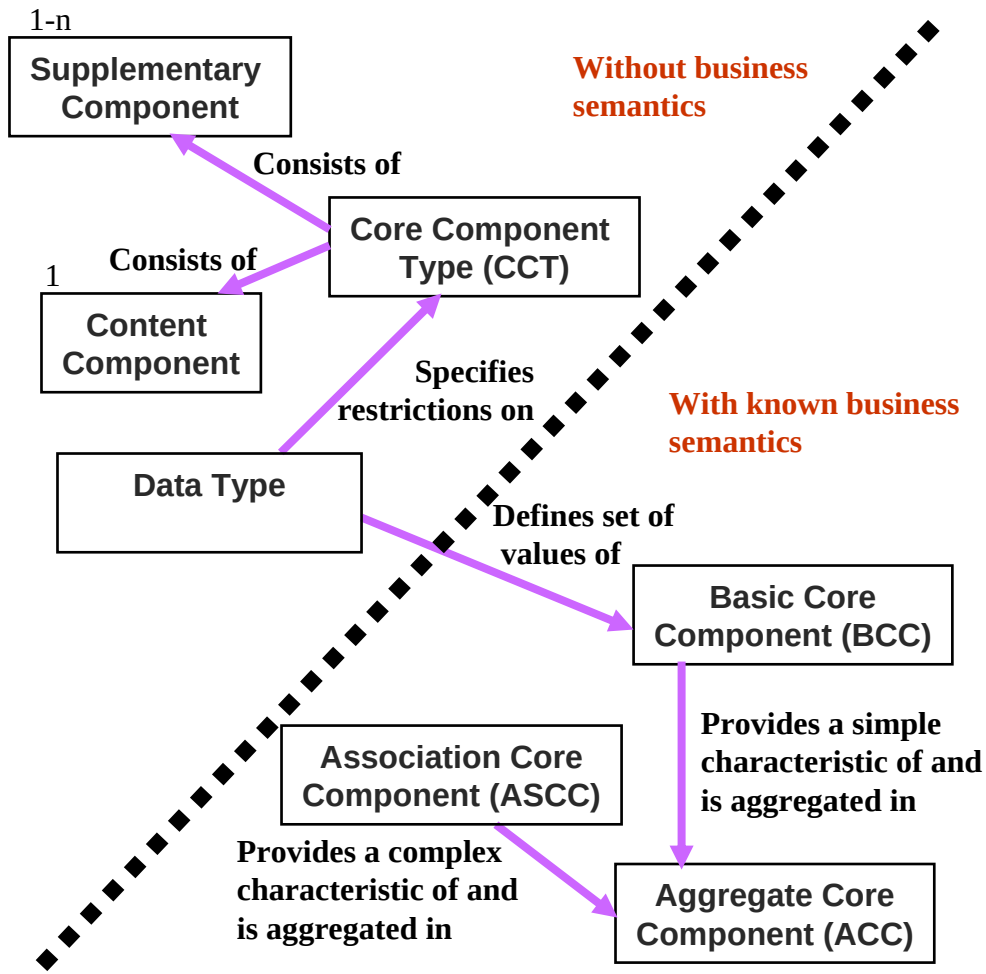
Key Concepts

- ISO 15000-5 provides the semantic, syntactic, lexical, and uniqueness rules called for in ISO 11179-5
- Approach is more flexible than current standards in this area because the semantic standardization is done in a syntax-neutral fashion
- Two trading partners using different syntaxes [e.g. XML and EDI are using *Business Semantics* in the same way]
- *Common Core Components underpinnings* enable clean mapping between disparate databases and message definitions across syntaxes, industry, and regional boundaries

Core Component Overview

Four categories of Core Components:

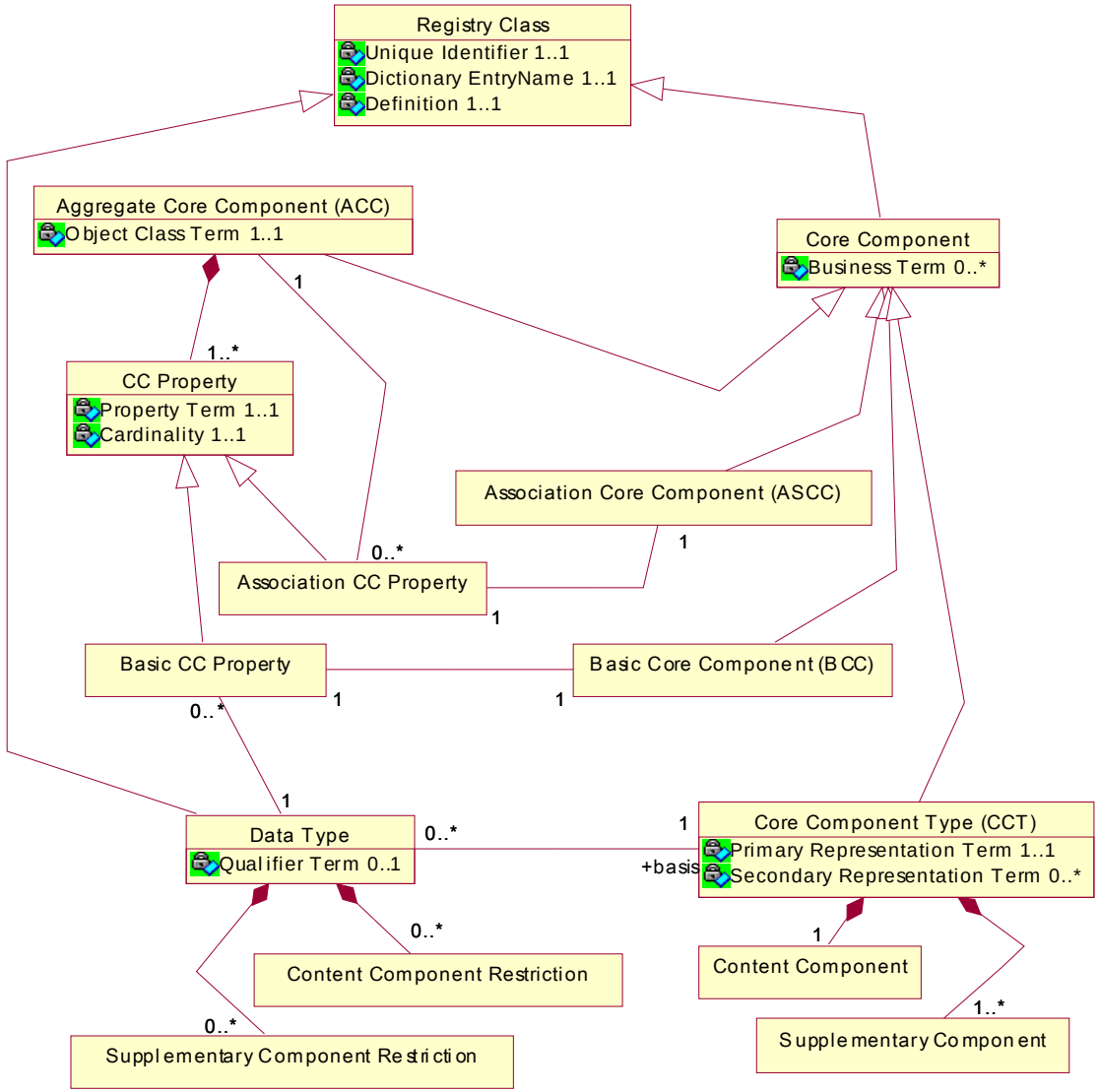
- Core Component Type (CCT)
- Basic Core Component (BCC)
- Aggregate Core Component (ACC)
- Association Core Component (ASCC)



ISO 15000-5

©Mark Crawford, LMI Government Consulting, April 2005

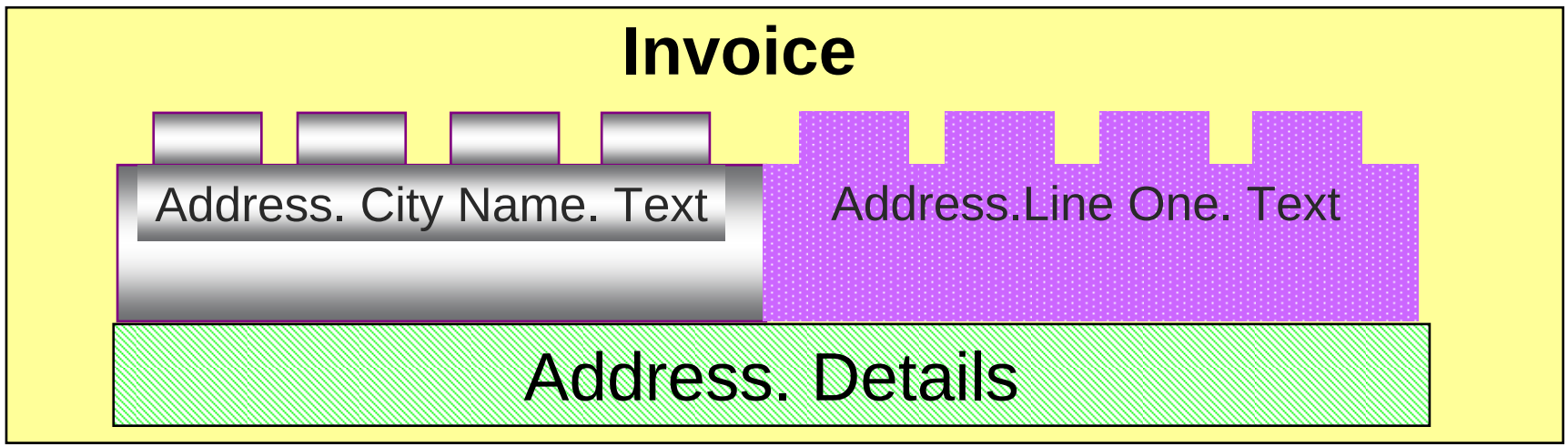
Technical Details CC and Data Types Metamodel



©Mark Crawford, LMI Government Consulting, April 2005

Core Component (CC) Definition

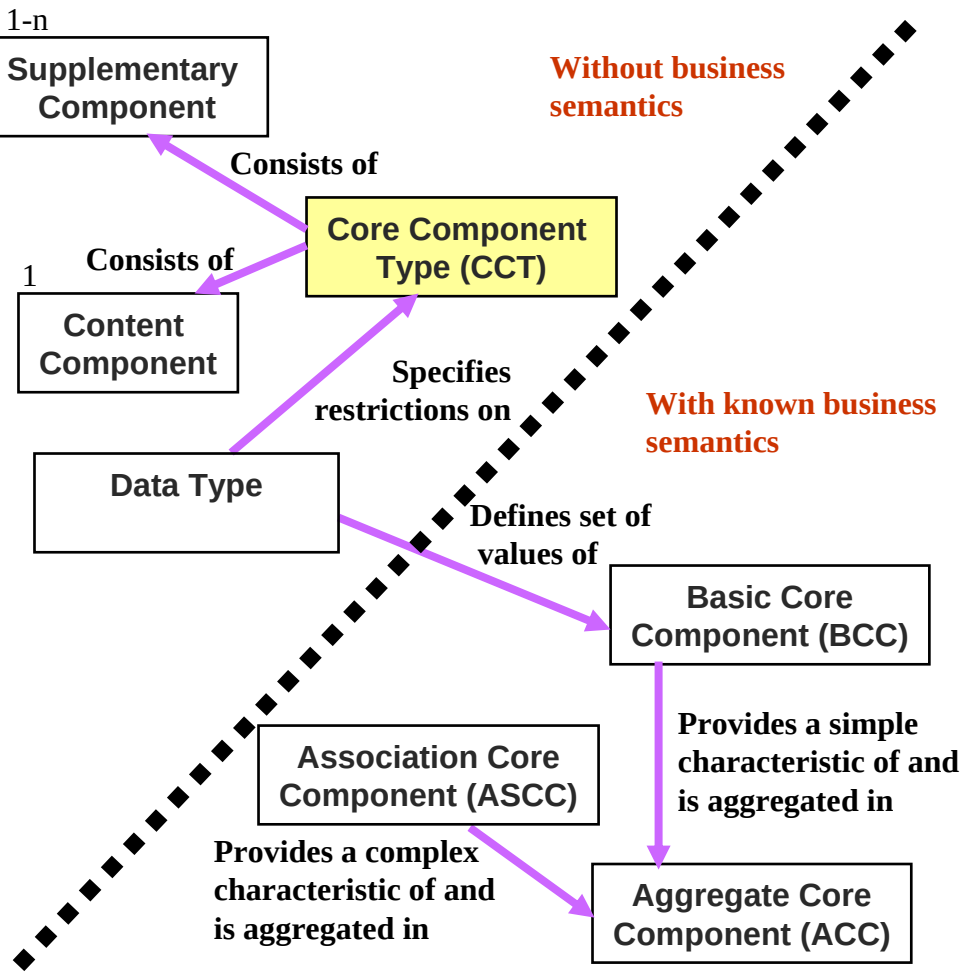
- A building block for the creation of a semantically correct and meaningful information exchange package
- Known as Core Components (CCs)
- Contains only the information pieces necessary to describe a specific concept
- Basis to construct all electronic business messages
- Basis for Business Information Entities



©Mark Crawford, LMI Government Consulting, April 2005

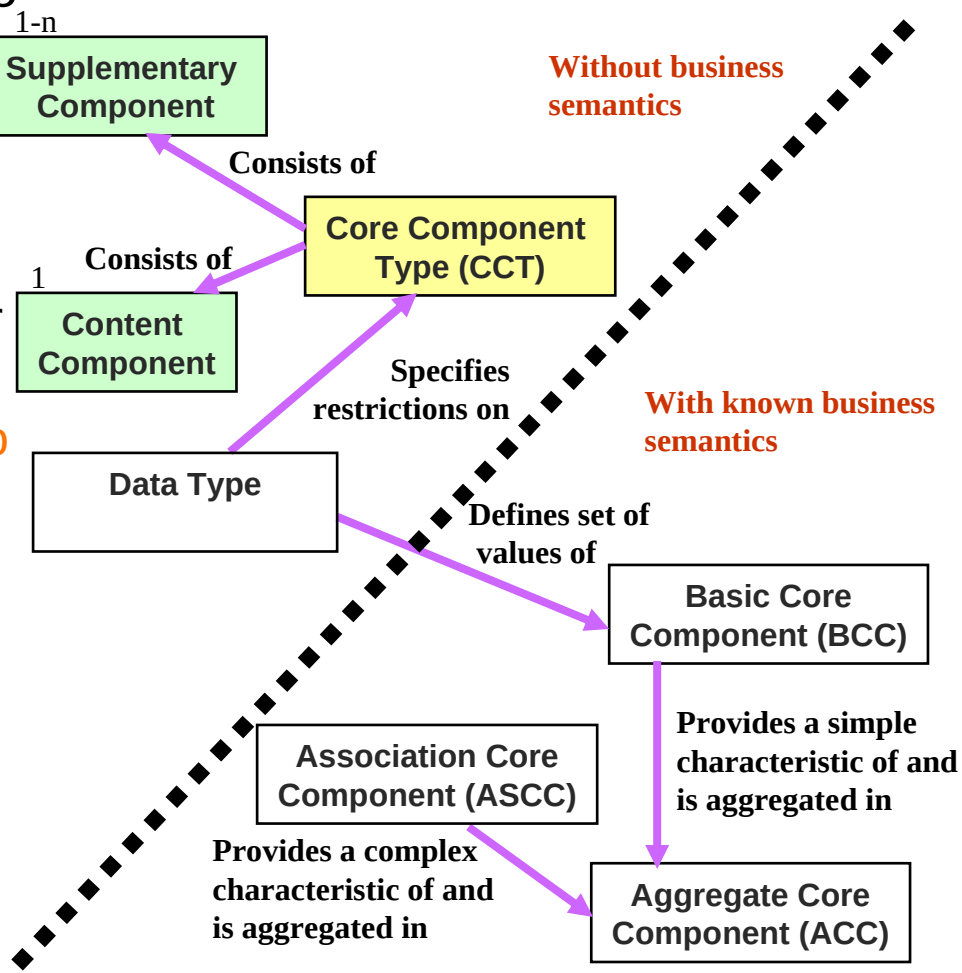
Core Component Type (CCT) Definition

- A Core Component, which consists of the actual data content plus one or more Supplementary Components that give essential extra definition to the Content Component
- Does not have Business Semantics
- Example: CCT for a specific amount of currency:
  - Amount.Type



Content Component and Supplementary Component

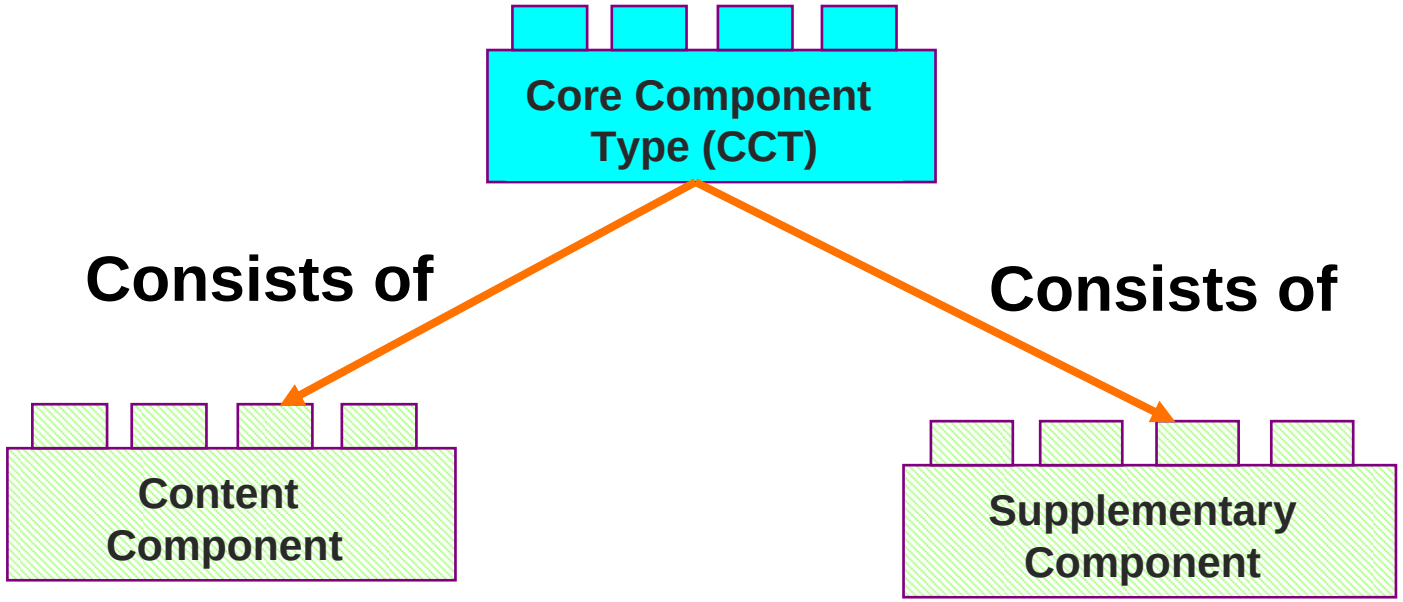
- Core Component Type = 12 Euro
- Content Component – defines the Primitive Type used to express the content of a Core Component Type
  - Example content component or Data value: 12
  - This content component has no semantic meaning on its own
- Supplementary Component – gives additional meaning to the Content Component in the Core Component Type
  - Example: Euro
  - Gives the essential extra definition/semantic meaning to the content component



©Mark Crawford, LMI Government Consulting, April 2005

Core Component Type (CCT) Example

LMI



Core Component Type: Measure. Type

Content Component: 15.45 ← Value has no semantic meaning

Supplementary Component: Inches ← Essential extra definition

15.45 Inches

©Mark Crawford, LMI Government Consulting, April 2005

Core Component Types

- [C7] The Core Component Type shall be one of the approved Core Component Types
  - **The approved core component types are contained in Table 8-1 in ISO 15000-5**
- **Amount. Type**
- **Binary Object. Type**
- **Code. Type**
- **Date Time. Type**
- **Identifier. Type**
- **Indicator. Type**
- **Measure. Type**
- **Numeric. Type**
- **Quantity. Type**
- **Text. Type**

Core Component Type

- [C8] The Content Component shall be the approved Content Component for the related Core Component Type
  - **The approved content component types are contained in Table 8-2 in ISO 15000-5**
- **Identifier. Content**
- **Code. Content**
- [C9] The Supplementary Component shall be one of the approved Supplementary Components for the related Core Component Type
  - **The approved supplementary component types are contained in Table 8-2 in ISO 15000-5**
- **Code List. Agency. Identifier**
- **Date Time. Format. Text**
- **Identification Scheme. Version. Identifier**
- **Measure Unit. Code**

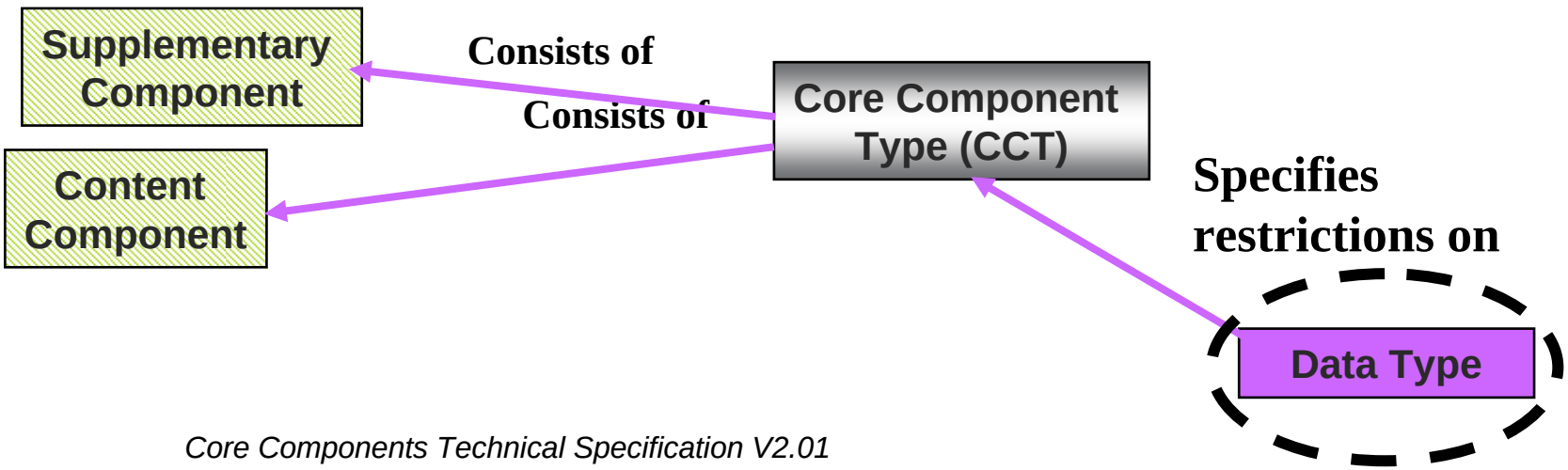
©Mark Crawford, LMI Government Consulting, April 2005

CCTS Definition of Data Types

LMI

6.1.2 Data Types

A *Data Type* defines the set of valid values that can be used for a particular ***Basic Core Component Property*** or ***Basic Business Information Entity Property***. It is defined by specifying restrictions on the ***Core Component Type*** from which the *Data Type* is derived. Figure 6-1 describes the *Data Type* and shows relationships to the *Core Component Type*.



Core Components Technical Specification V2.01

Part 8 of the ebXML Framework

CCTS Data Type Rule #1

LMI

[D1] A Data Type shall be based on one of the approved Core Component Types.

- Some Core Component Types have more than one representation term (See table 8-3 in CCTS)
- This means that there are more data types than core component types

©Mark Crawford, LMI Government Consulting, April 2005

List of Permissible Representation Terms

Representation Term	Related CCT	Secondary Rep Term
Amount	Amount. Type	
Binary Object	Binary Object. Type	Graphic, Picture, Sound, Video
Code	Code. Type	
Date Time	Date Time. Type	Date, Time
Identifier	Identifier. Type	
Indicator	Indicator. Type	
Measure	Measure. Type	
Numeric	Numeric. Type	Value, Rate, Percent
Quantity	Quantity. Type	
Text	Text. Type	Name

Table 8-3 CCTS V1.9

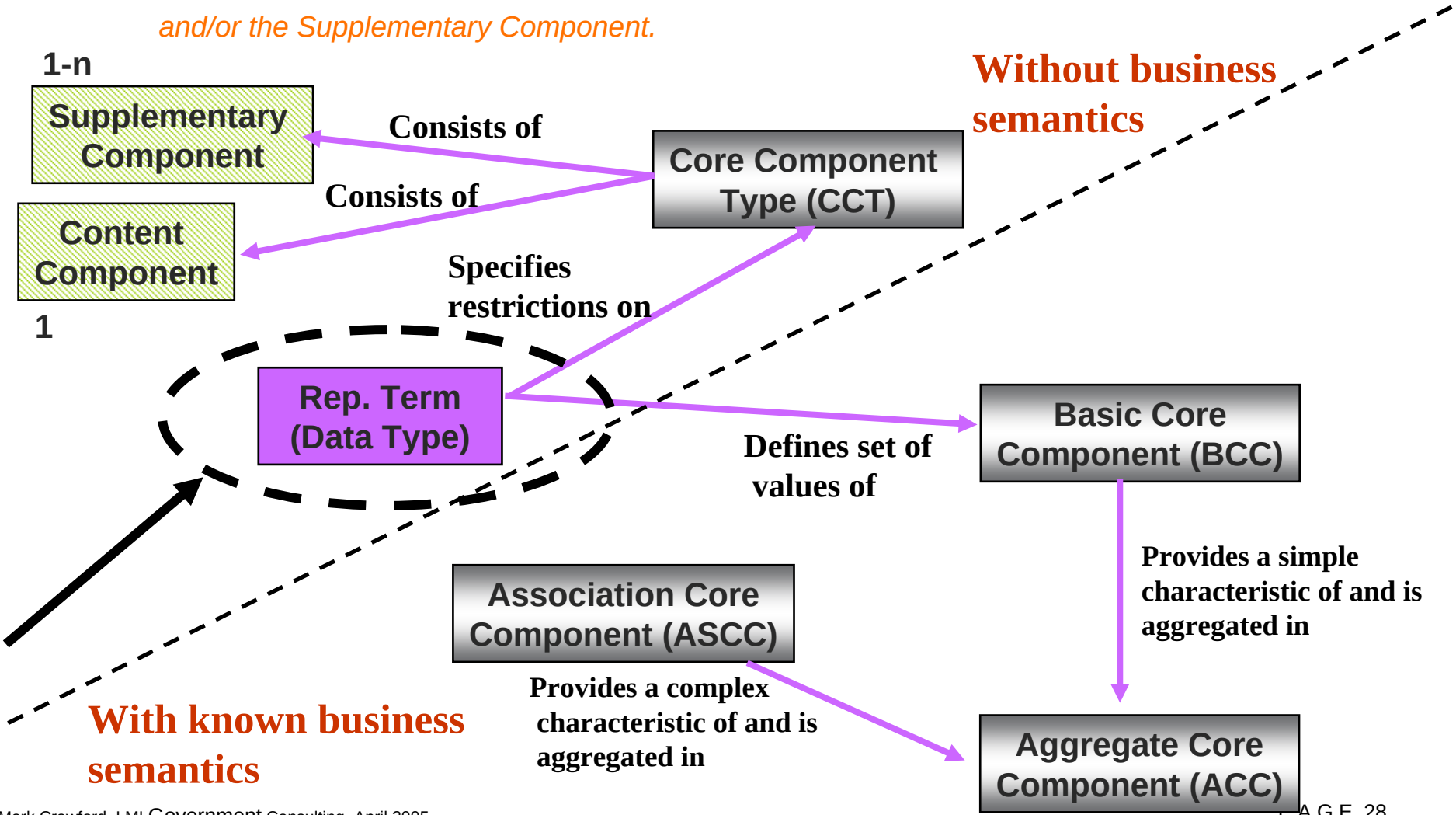
Core Components Technical Specification V2.01

Part 8 of the ebXML Framework

©Mark Crawford, LMI Government Consulting, April 2005

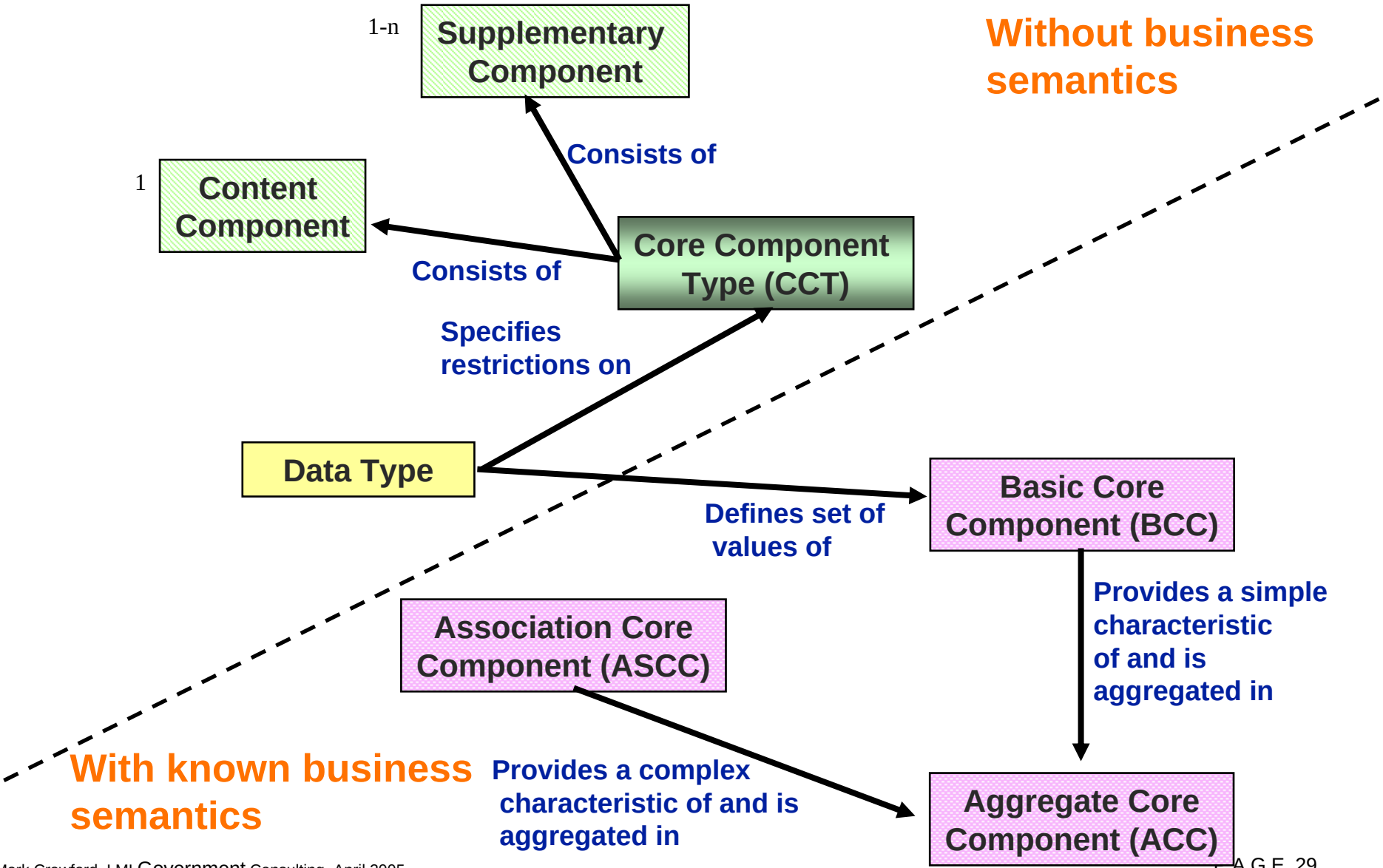
CCTS Data Types Rule #2

[D2] *Where necessary, a Data Type shall restrict the set of valid values allowed by the Core Component Type on which it is based, by imposing restrictions on the Content Component and/or the Supplementary Component.*



©Mark Crawford, LMI Government Consulting, April 2005

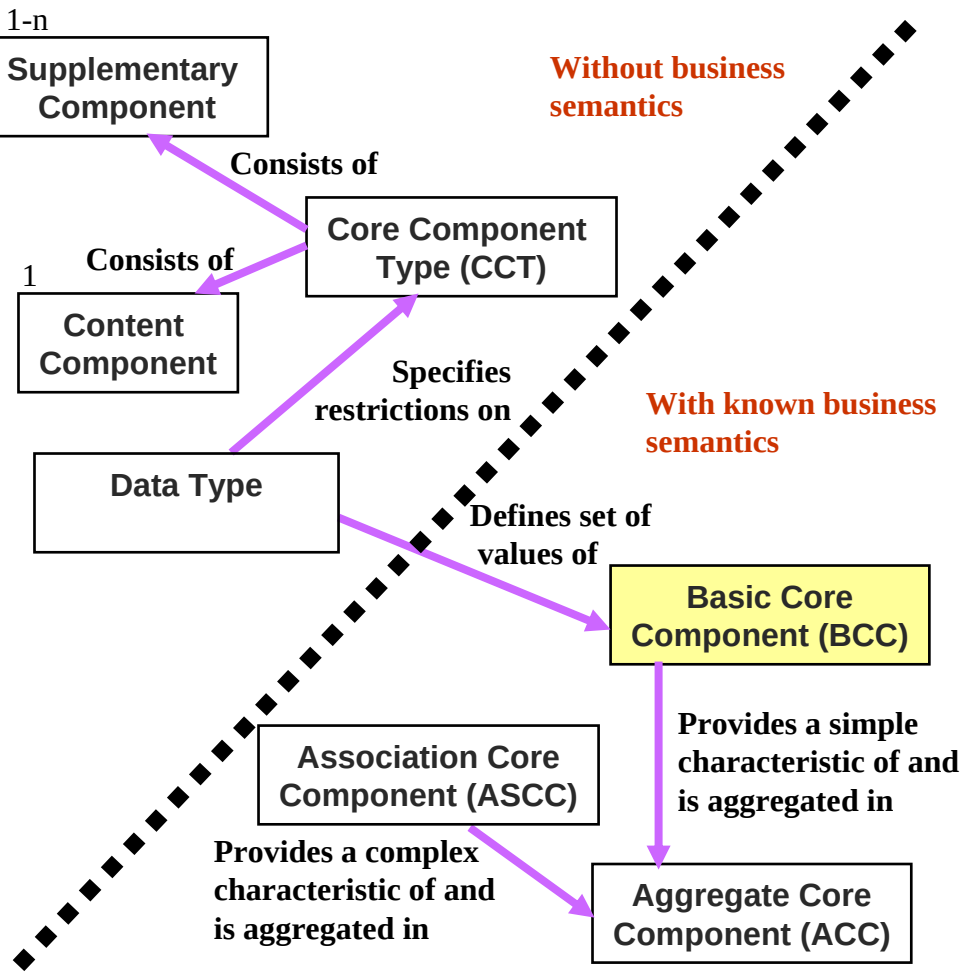
Summary: Core Component Constructs



©Mark Crawford, LMI Government Consulting, April 2005

Basic Core Component (BCC) Definition

- A Core Component which constitutes a singular business characteristic of a specific Aggregate Core Component that represents an Object Class
- It has a unique Business Semantic definition
- Represents a Basic Core Component Property and is therefore of a Data Type, which defines its set of values
- Function as the Properties of Aggregate Core Components

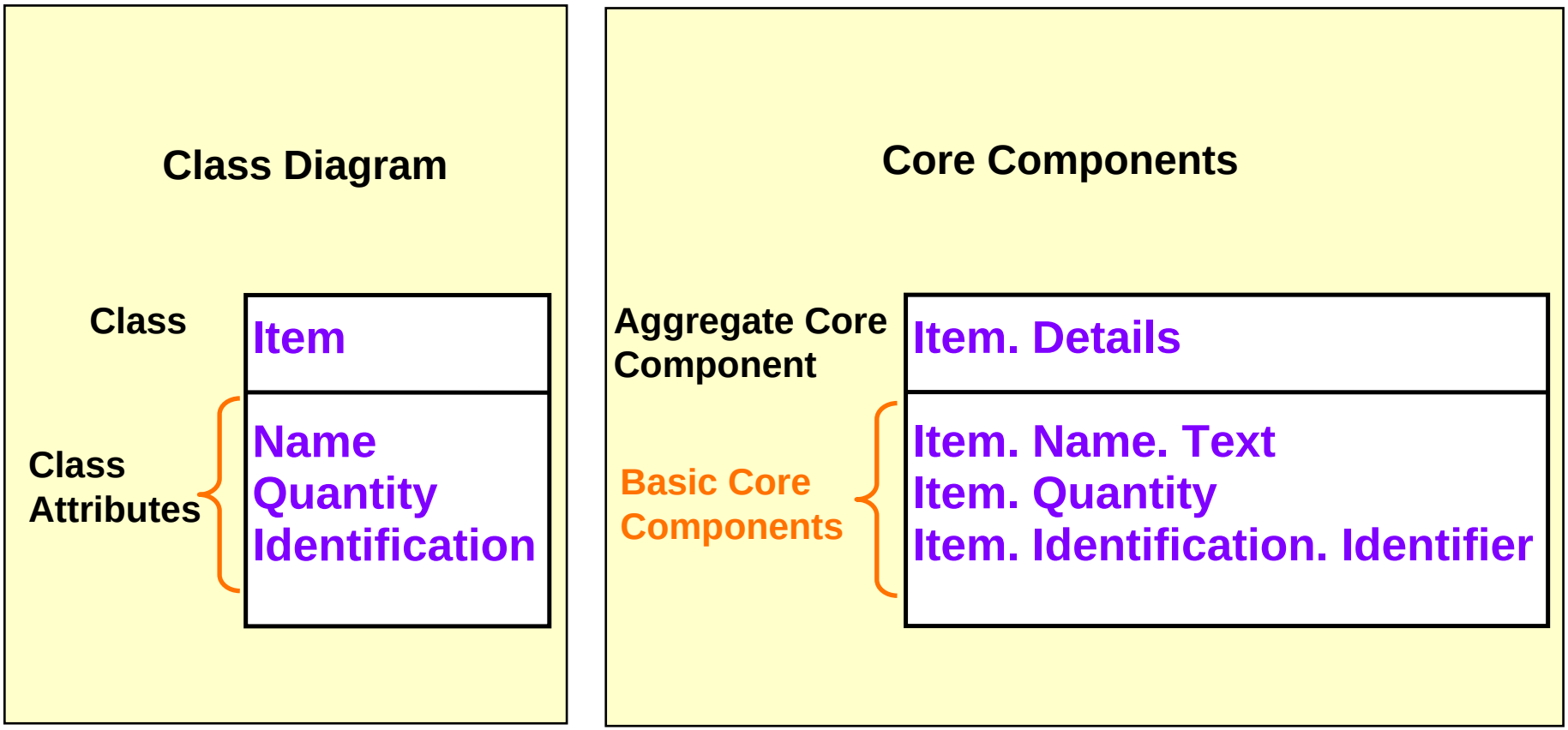


Basic Core Component Example

- Item. Name. Text
- Organization. Name. Text
- Organization. Description. Text
- Address. Street. Text
- Address. City Name. Text

Basic Core Component (BCC)

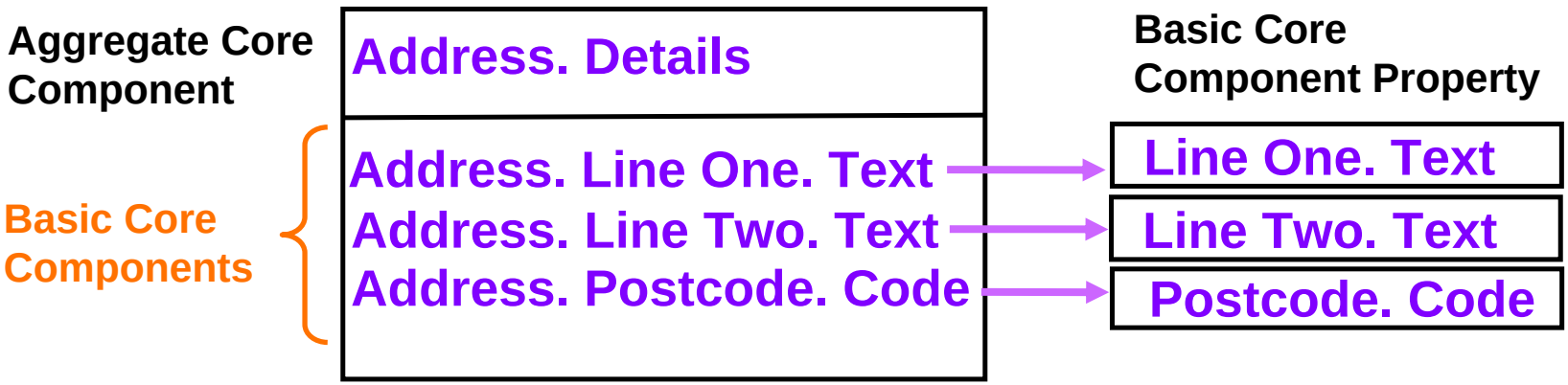
- Property of an Aggregate Core Component
  - Attribute of a class



©Mark Crawford, LMI Government Consulting, April 2005

Basic Core Component (BCC) Example

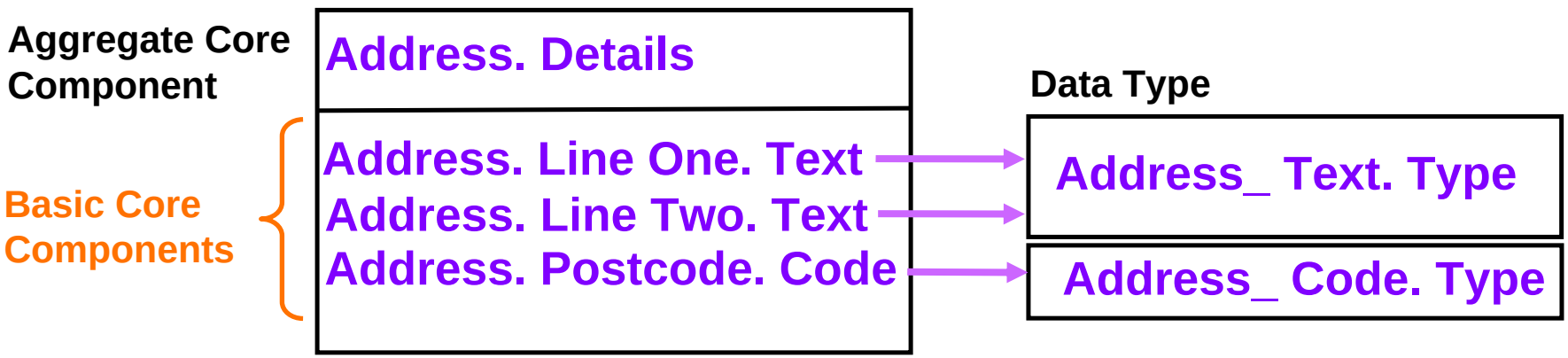
- Has a Basic Core Component Property



©Mark Crawford, LMI Government Consulting, April 2005

Basic Core Component (BCC) Example

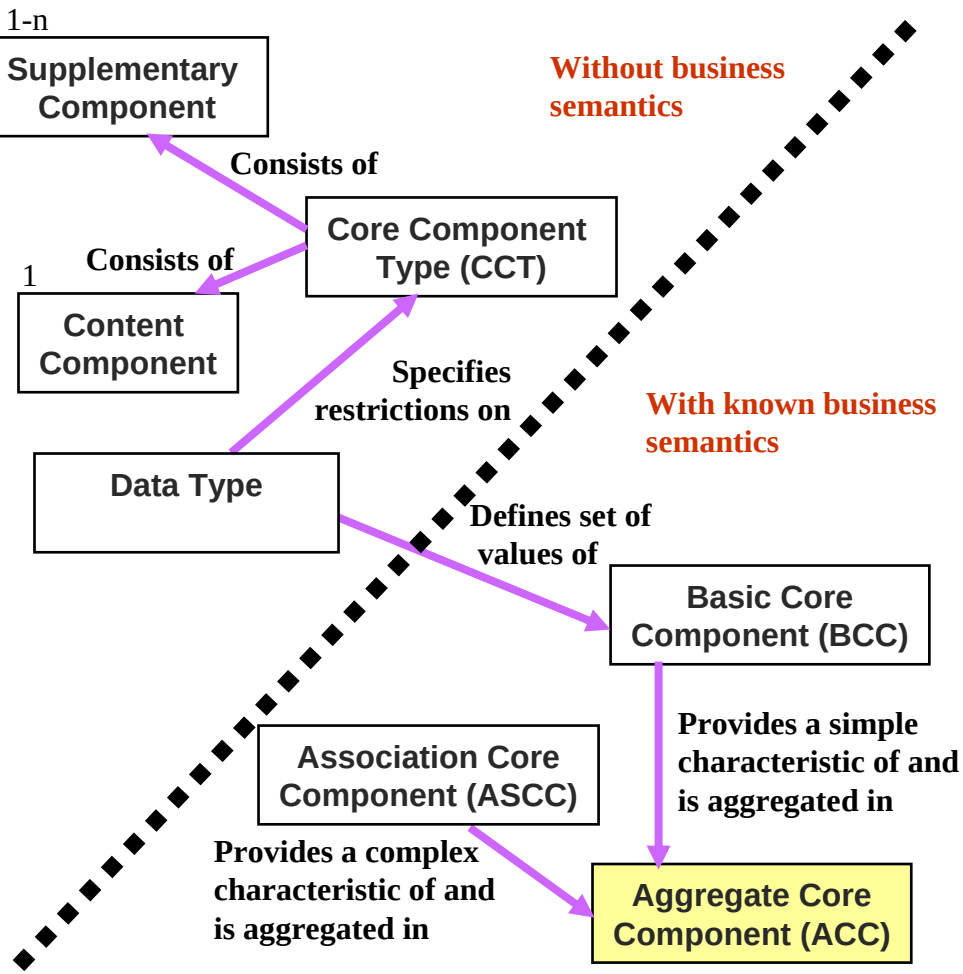
- Has a Data Type



©Mark Crawford, LMI Government Consulting, April 2005

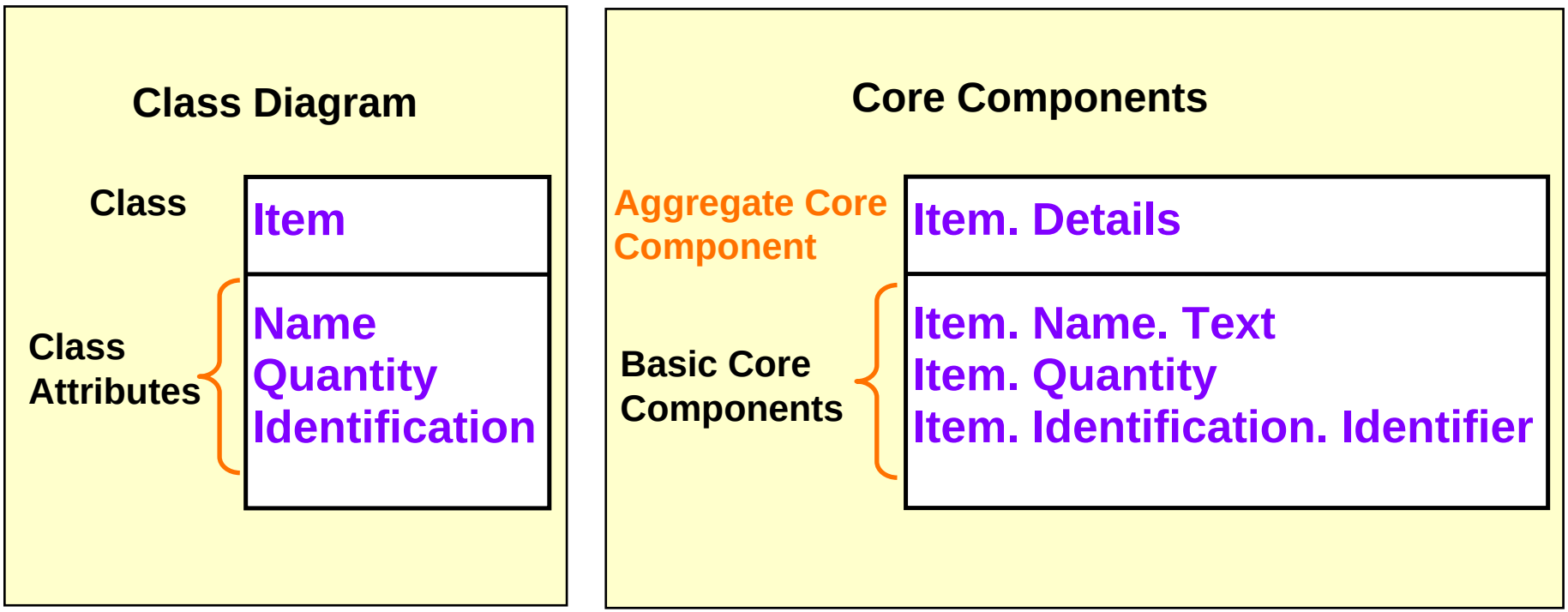
Aggregate Core Component (ACC) Definition

- A collection of related pieces of business information that together convey a distinct business meaning
- Independent of any specific Business Context
- Expressed in modeling terms, it is the representation of an Object Class



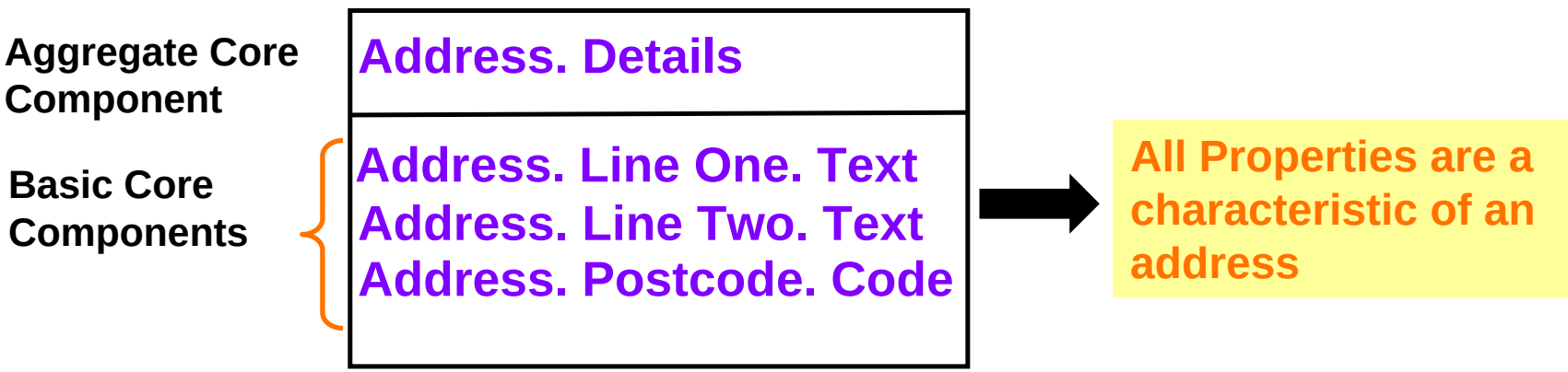
Aggregate Core Component (ACC)

- Representation of an Object Class
- In a real business circumstance serves as the basis of an Aggregate Business Information Entity



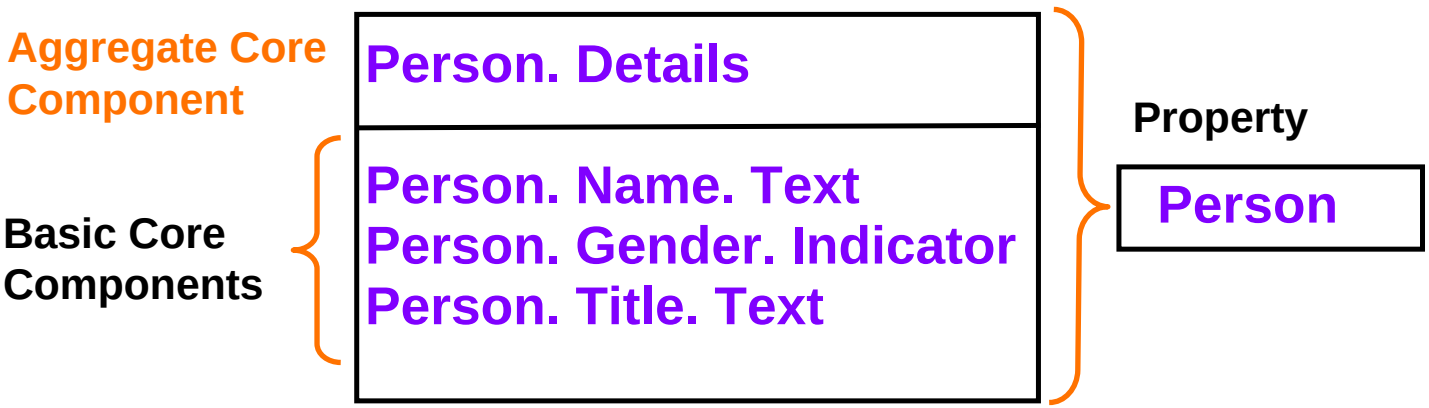
Aggregate Core Component (ACC) Rules

- [C2] Within an Aggregate Core Component, all embedded Core Component Properties shall be related to the concept of the aggregate property
  - Example:



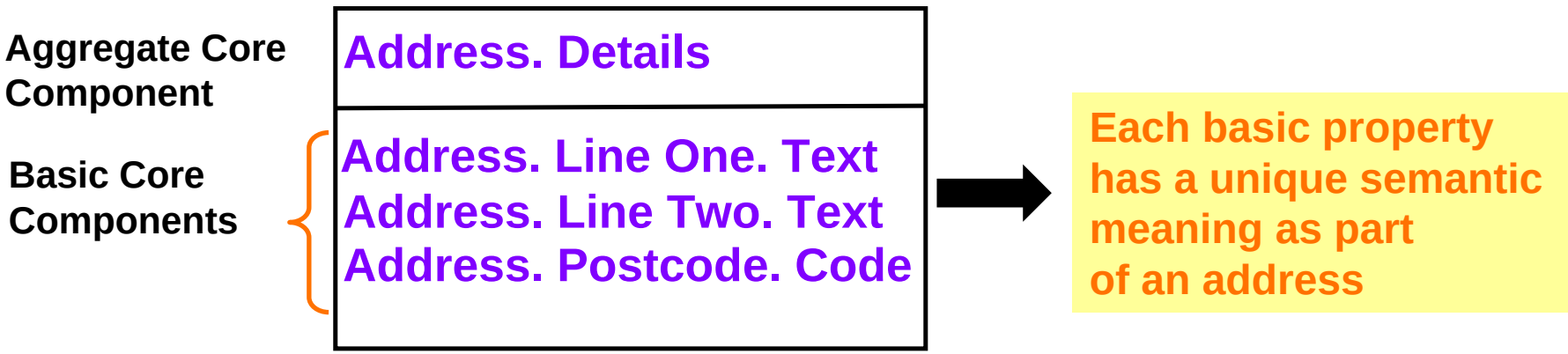
Aggregate Core Component (ACC)

- Has a Core Component Property that defines the business characteristic



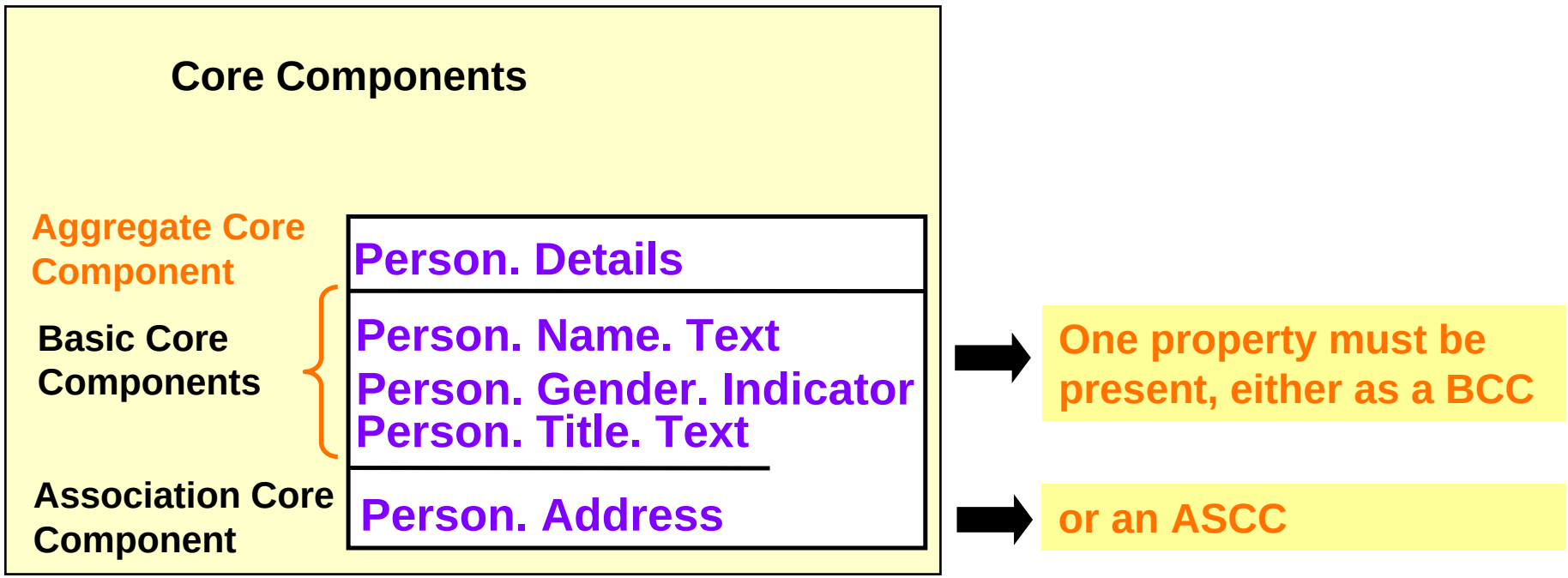
Aggregate Core Component (ACC) Rules

- [C3] There shall be no semantic overlap between the Core Component Properties embedded within the same Aggregate Core Component
  - Example:



Aggregate Core Component (ACC) Rules

- [C5] An Aggregate Core Component shall contain at least one Core Component Property. A Core Component Property shall be either a Basic Core Component Property or an Association Core Component Property.
  - Example:



Aggregate Core Component (ACC) Examples

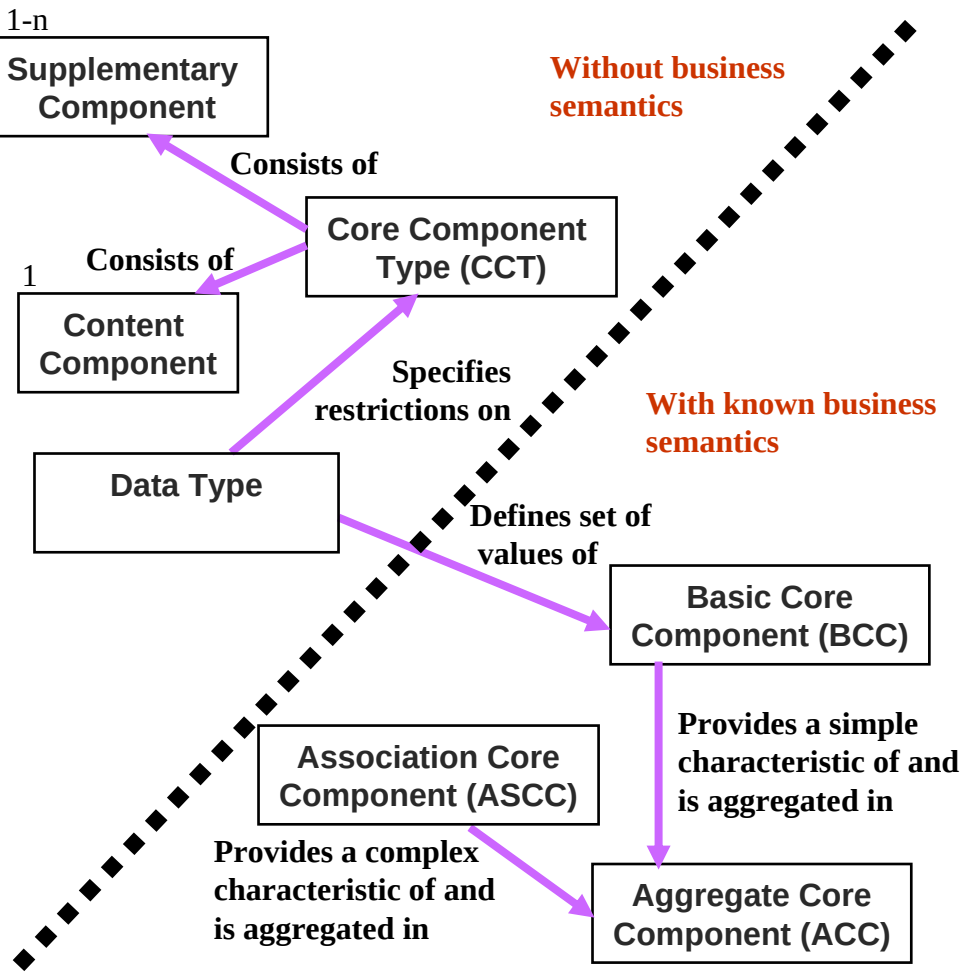
LMI

- **Contact. Details**
- **Delivery. Details**
- **Facility. Details**
- **Location. Details**
- **Organization. Details**
- **Party. Details**
- **Report. Details**

©Mark Crawford, LMI Government Consulting, April 2005

Association Core Component (ASCC) Definition

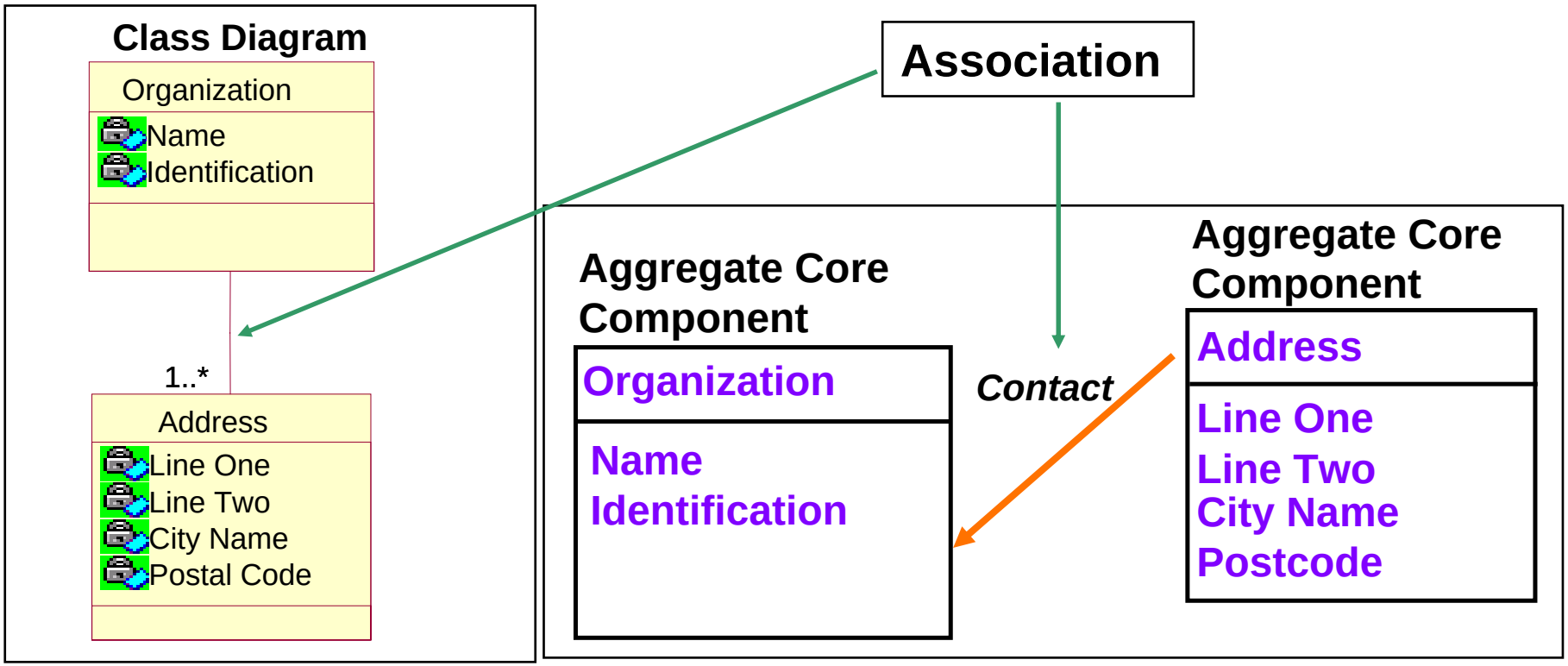
- A Core Component which constitutes a complex business characteristic of a specific Aggregate Core Component that represents an Object Class
- It has a unique Business Semantic definition
- Represents an Association Core Component Property and is associated to an Aggregate Core Component, which describes its structure



©Mark Crawford, LMI Government Consulting, April 2005

Association Core Component (ASCC)

- An ASCC is a Core Component naming mechanism for expressing the relationship between two object classes
  - Object Oriented inheritance that retains semantic clarity that can not be expressed in UMM
  - Expresses the structure of the association

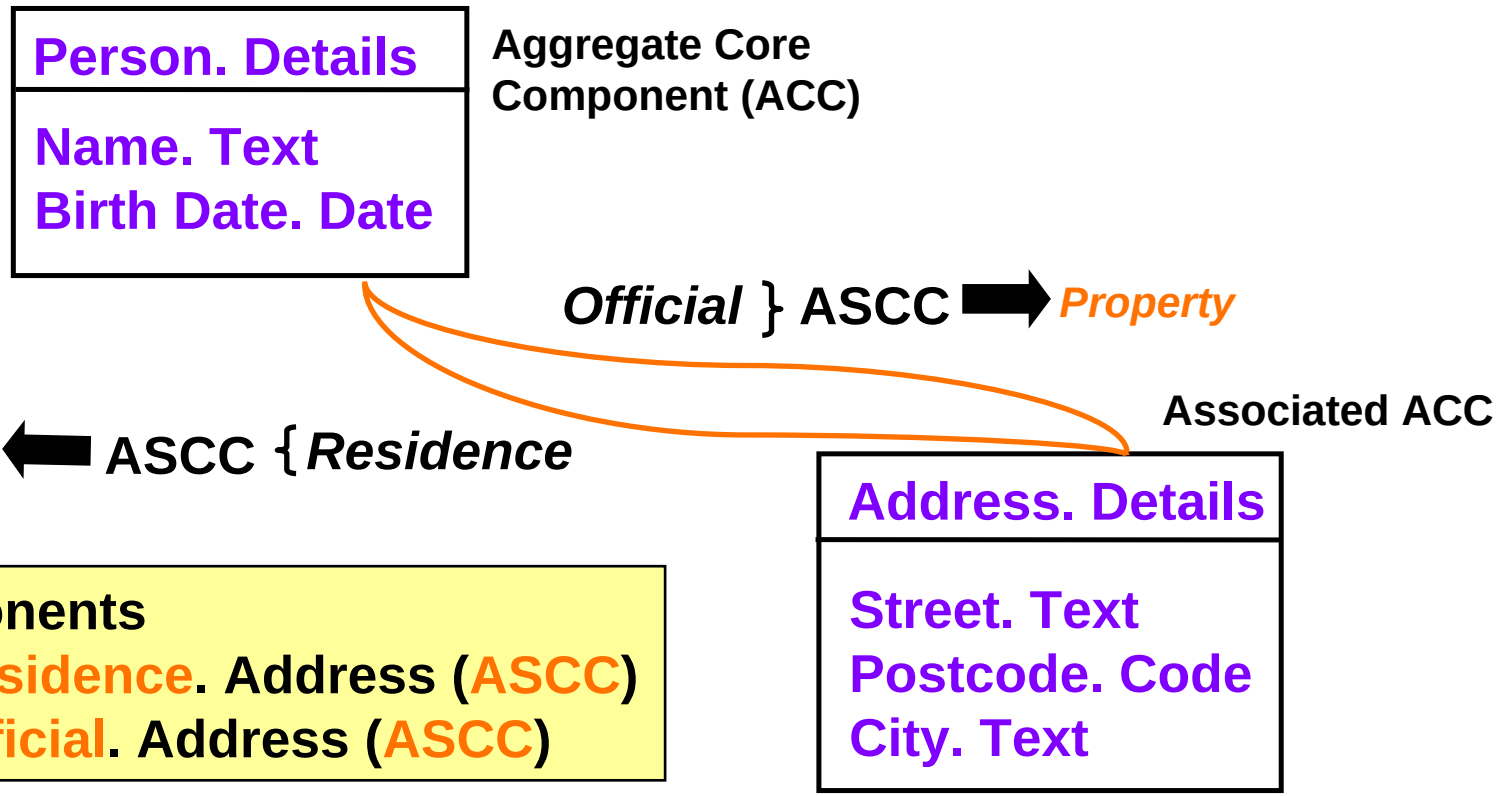


Association Core Component (ASCC) Example

LMI

ASCC=

- Object Class Term of the ACC that contains the ASCC (Person)
- Property Term that represents the property of the ASCC (Official/Residence)
- Object Class Term of the ACC that describes the structure of the ASCC (Address)

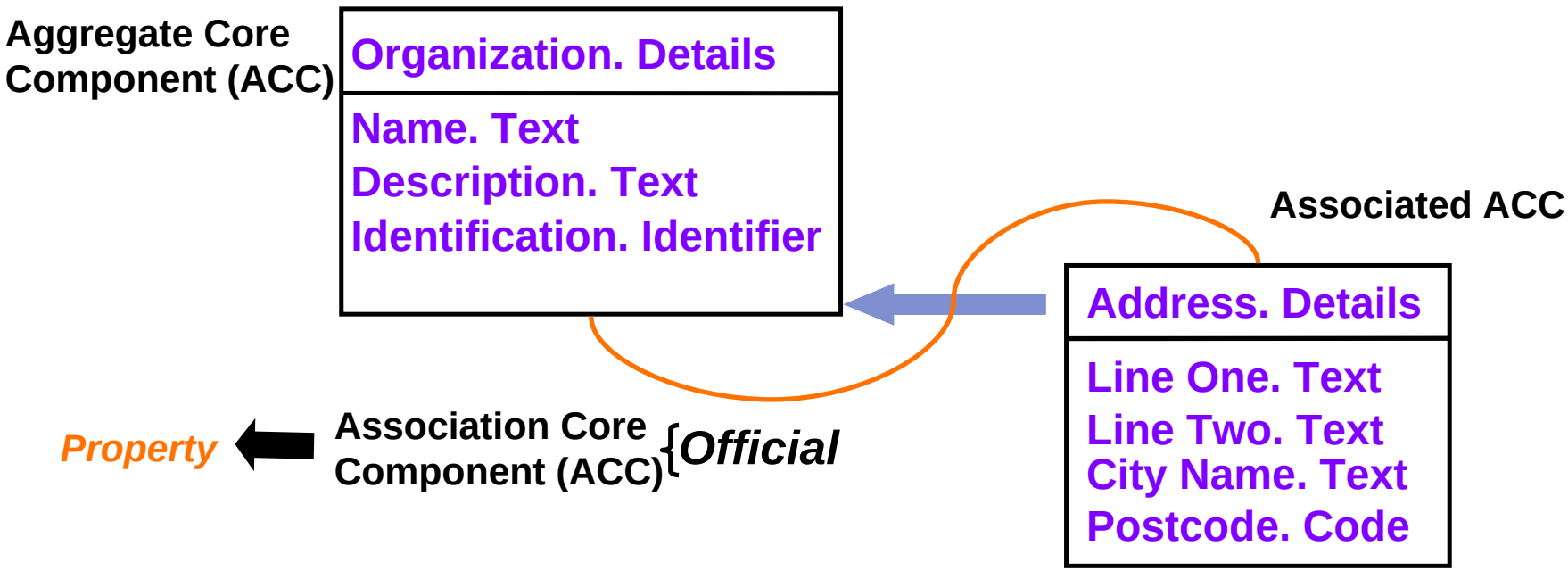


Core Components
- Person. Residence. Address (ASCC)
- Person. Official. Address (ASCC)

©Mark Crawford, LMI Government Consulting, April 2005

Association Core Component (ASCC)

- Expressing the structure of the association

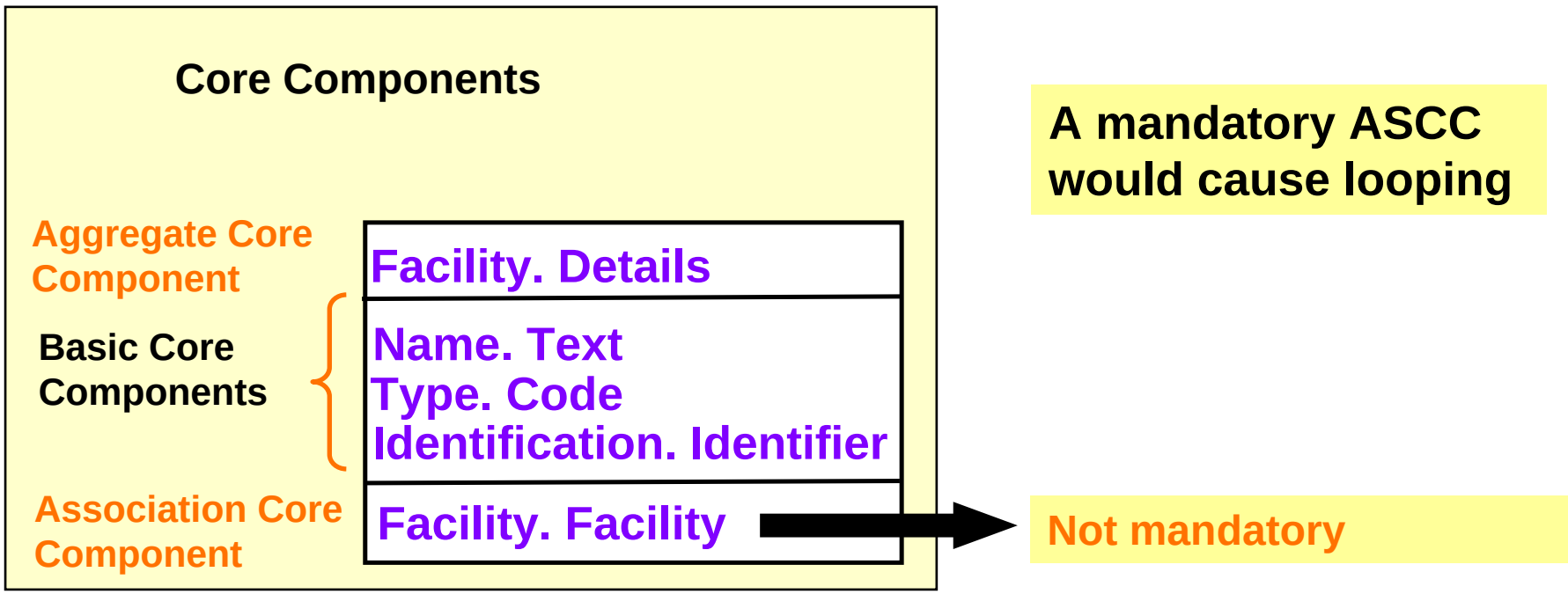


The structure of the Association Core Component is described by Address. Details

Aggregate Core Component (ACC) Nested Association Rule

LMI

- [C6] An Aggregate Core Component shall never contain – indirectly or at any nested level – a mandatory Association Core Component Property that references itself.
  - Example:

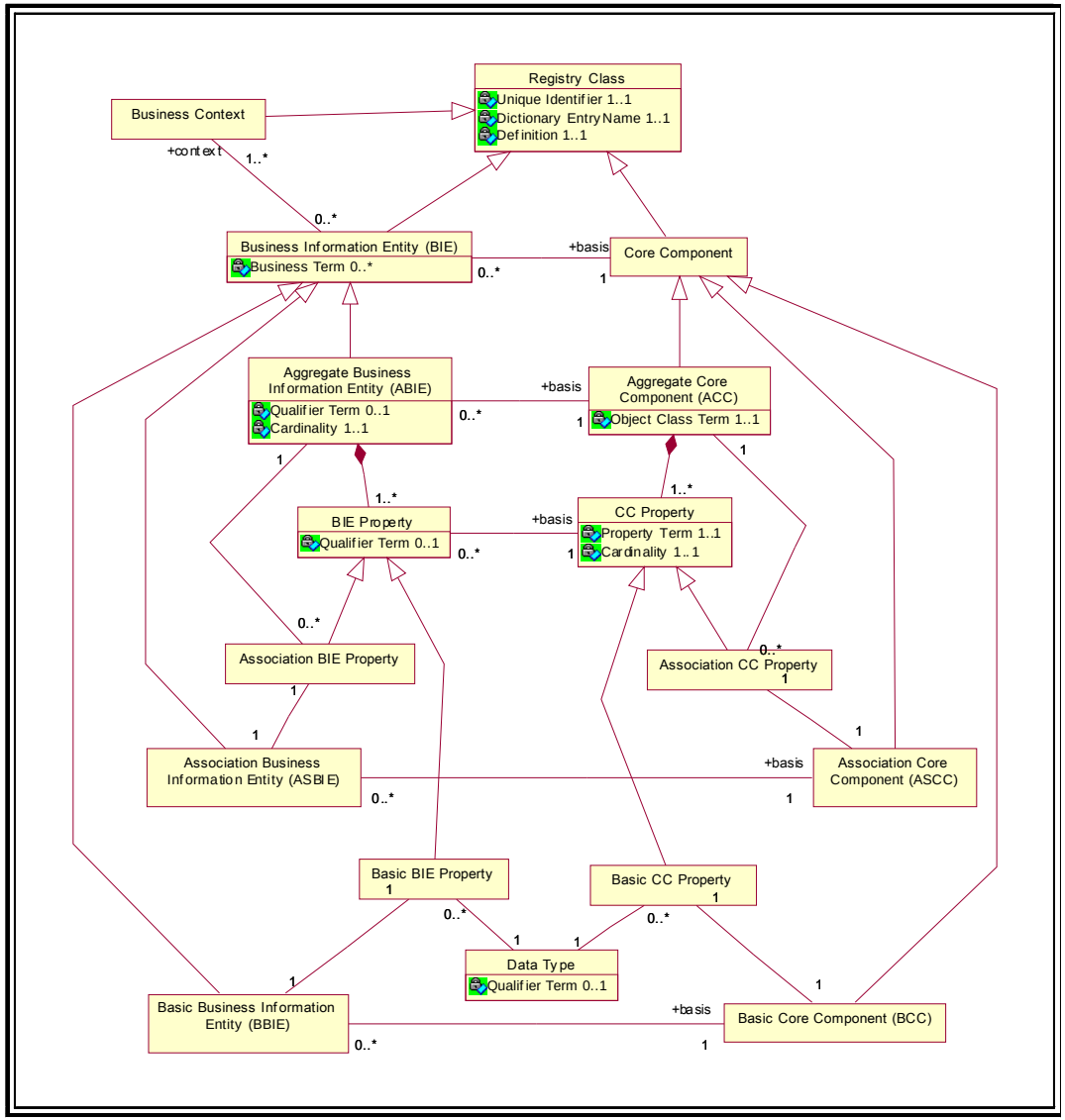


The Rest of the Story

LMI

- Core Components are the building blocks for Business Information Entities
- The key differentiator between Core Components and Business Information Entities is the concept of Business Context
- Business Context is a mechanism for qualifying and refining Core Components according to their use under particular business circumstances
- Once Business Contexts are identified, Core Components can be differentiated to take into account any necessary qualification and refinement needed to support the use of the Core Component in the given Business Context

Technical Details BIE Metamodel



Business Context Definition

LMI

- The formal description of a specific business circumstance as identified by the values of a set of Context Categories
  - Allows different business circumstances to be uniquely distinguished
- ISO 15000-5 identifies eight context categories
  - Business Process, Production Classification, Industry Classification, Geopolitical, Official Constraints, Business Process Role, Supporting Role, and System Capabilities

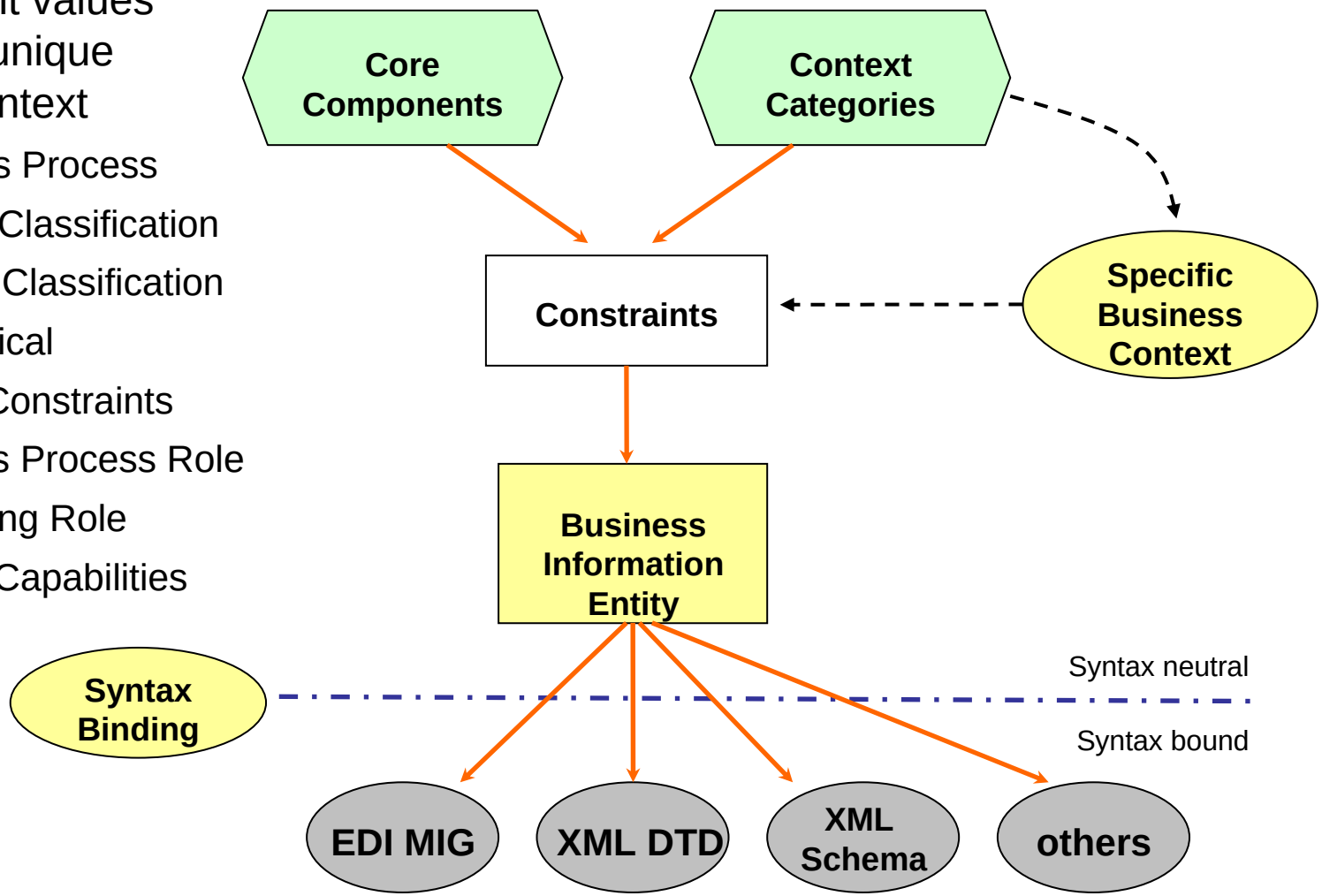
Example: Geopolitical Contexts – allow description of those aspects related to region, nationality, or geographically based cultural factors.

Global, Continent, Economic Region, Country

©Mark Crawford, LMI Government Consulting, April 2005

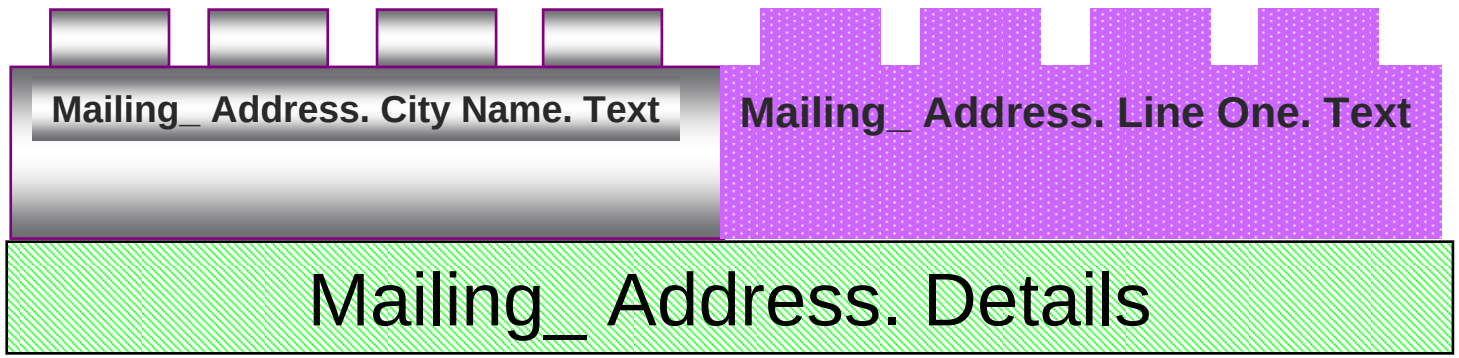
Context Application

- A set of eight values identifies a unique business context
  - Business Process
  - Product Classification
  - Industry Classification
  - Geopolitical
  - Official Constraints
  - Business Process Role
  - Supporting Role
  - System Capabilities



Business Information Entity (BIE) Definition

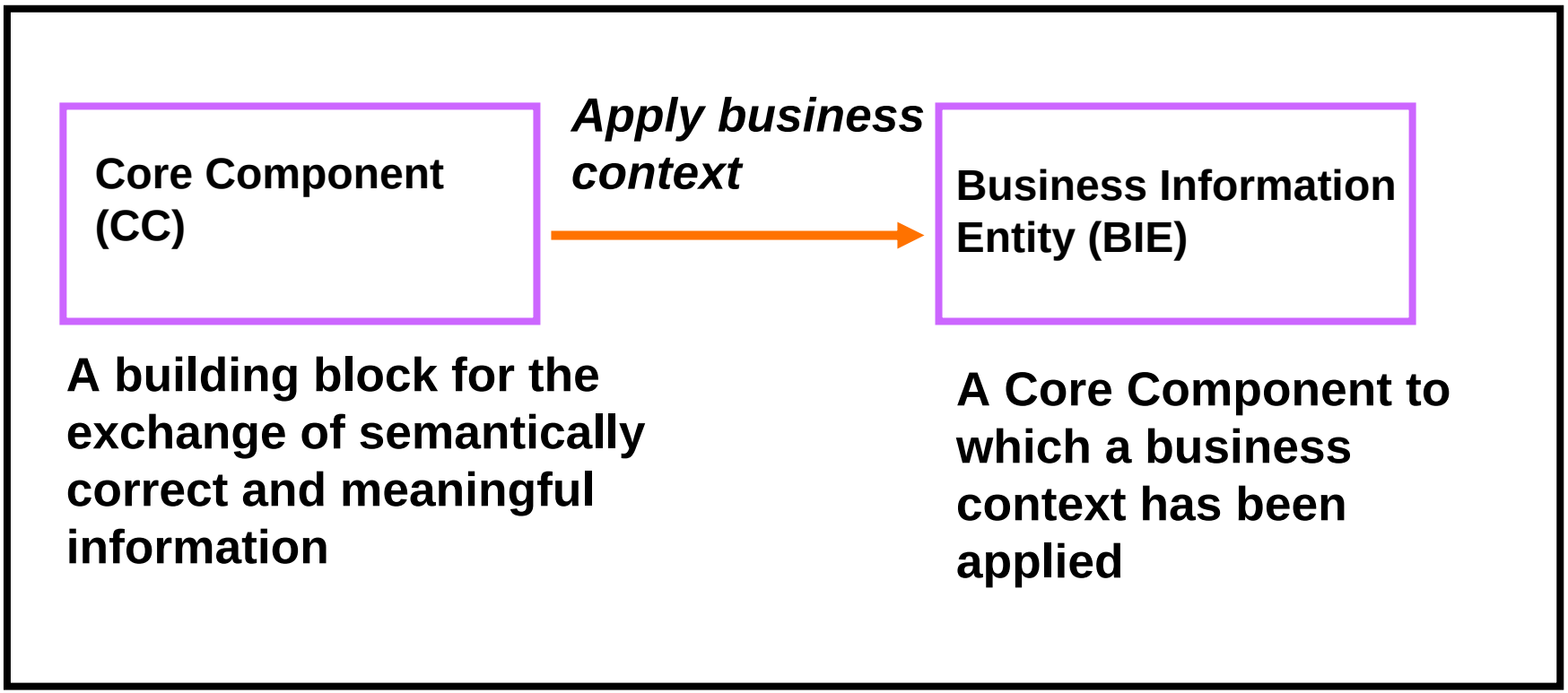
- A piece of business data or a group of pieces of business data with a unique Business Semantic definition.
- A Business Information Entity can be:
  - a Basic Business Information Entity (BBIE),
  - an Association Business Information Entity (ASBIE),
  - or an Aggregate Business Information Entity (ABIE).



©Mark Crawford, LMI Government Consulting, April 2005

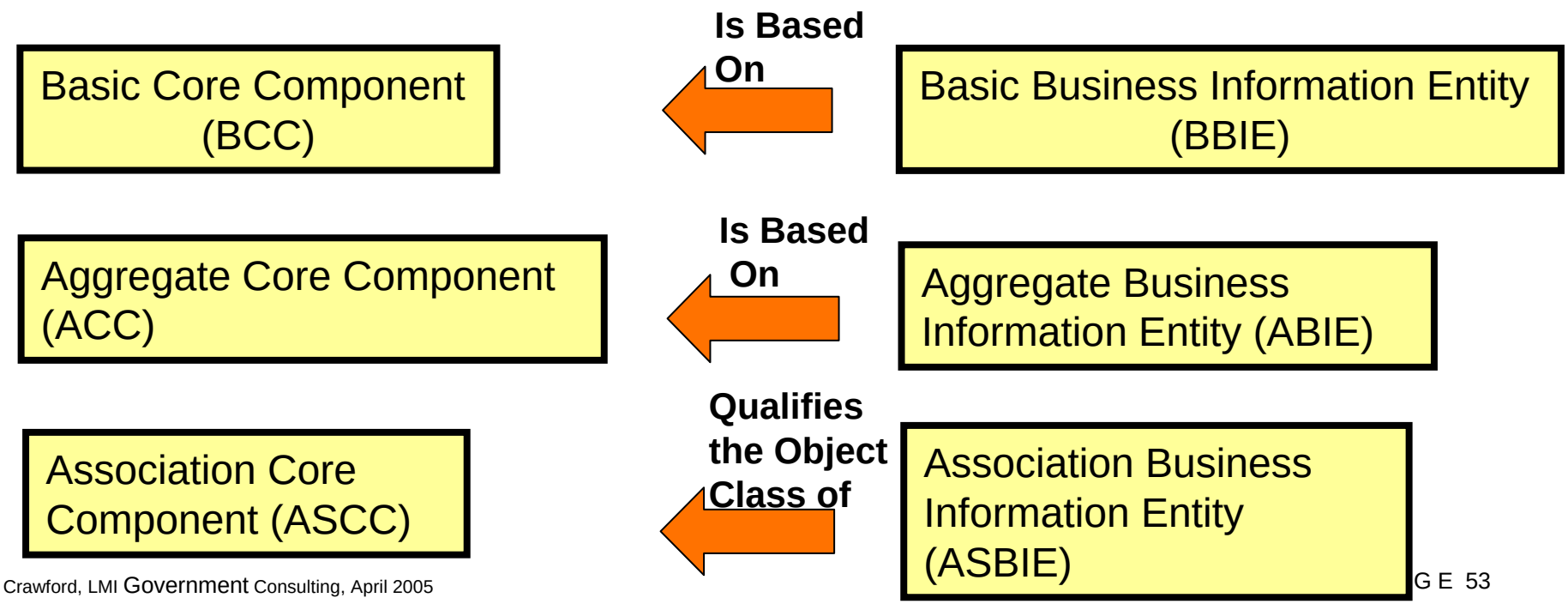
Business Information Entity (BIE)

- A Core Component used in a real business circumstance
- A Core Component with business context applied



Relationship of CCs and BIEs

- A Basic Business Information Entity is based on a Basic Core Component (BCC)
- An Aggregate Business Information Entity is a re-use of an Aggregate Core Component (ACC) in a specified Business Context
- An Association Business Information Entity is based on an Association Core Component (ASCC)



CC/BIE Relationship Example

LMI

Multiple ABIEs can be created from an ACC

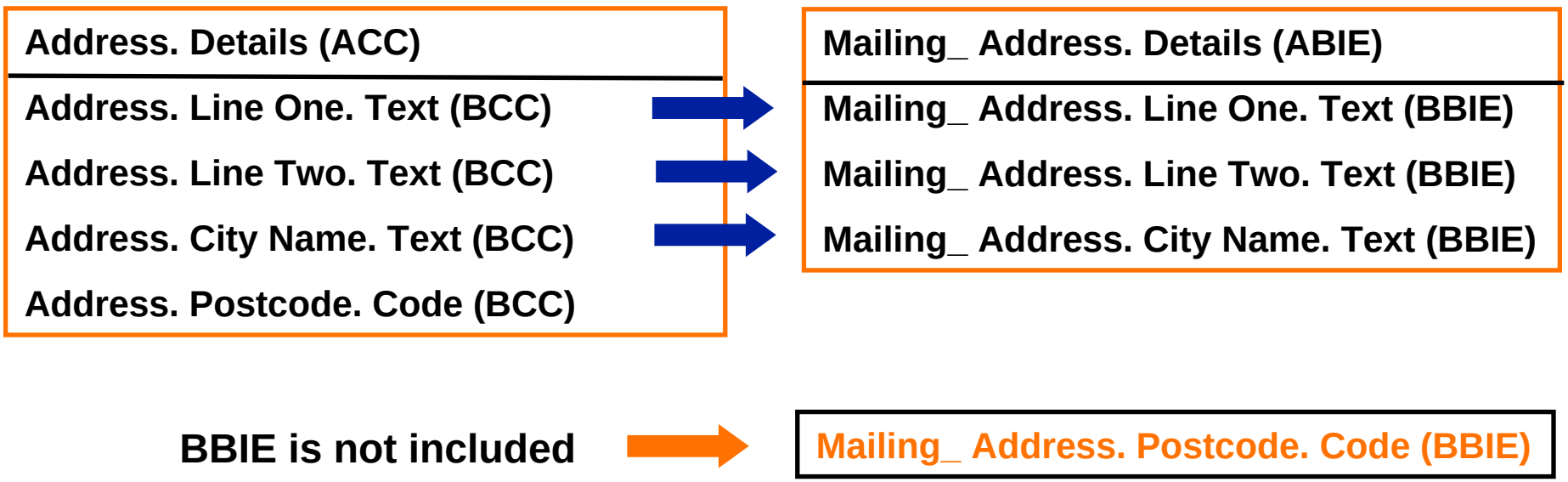
Multiple BBIEs can be created from a BCC

Core Components	Business Information Entities
Organization. Name. Text (BCC)	Supplier_ Organization. Name. Text (BBIE)
	Supplier_ Organization. Department_ Name. Text (BBIE)
Organization. Identification. Identifier (BCC)	Supplier_ Organization. Department_ Identification. Identifier (BBIE)
Address. Details (ACC)	Mailing_ Address. Details (ABIE)
	Shipping_ Address. Details (ABIE)

©Mark Crawford, LMI Government Consulting, April 2005

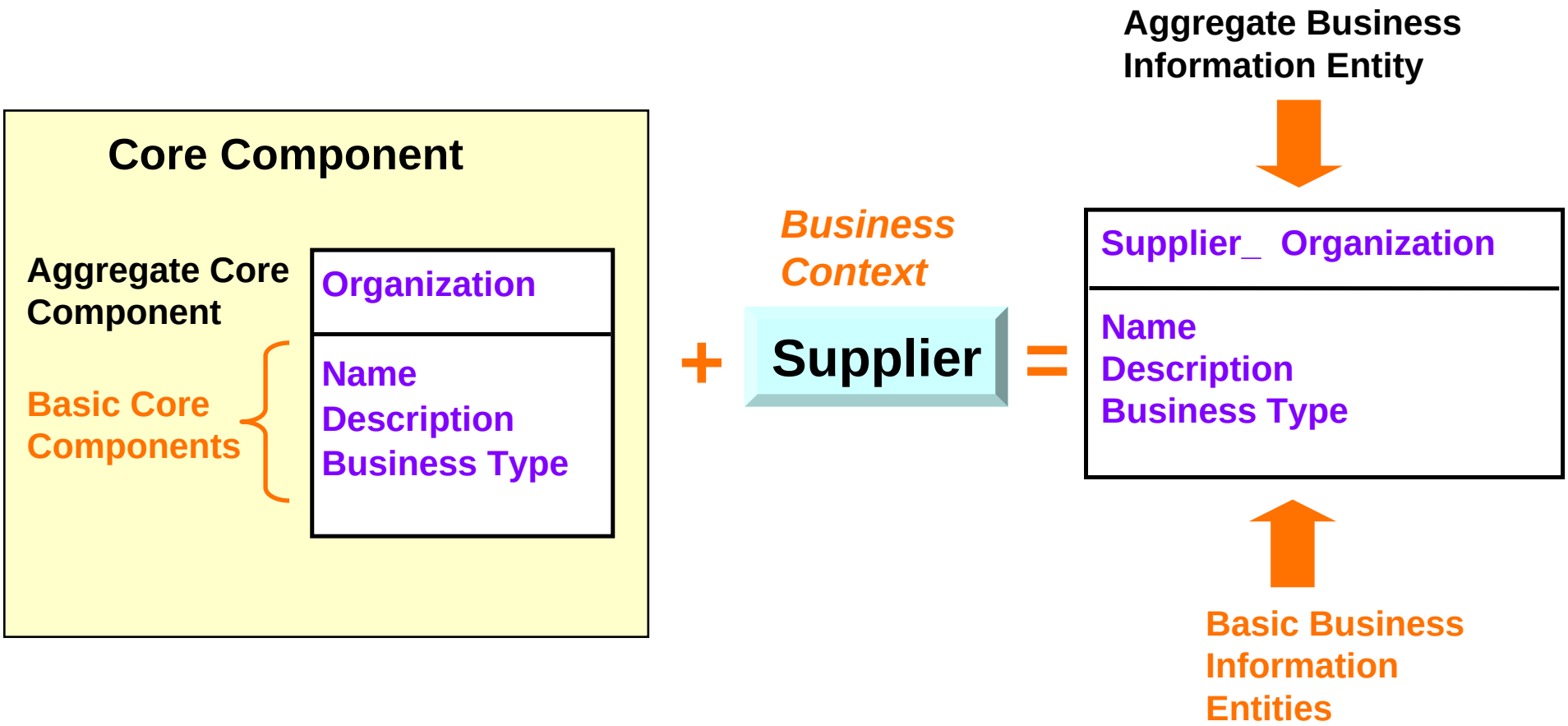
An ACC can be restricted

An ABIE does not need to include all attributes (BBIEs)



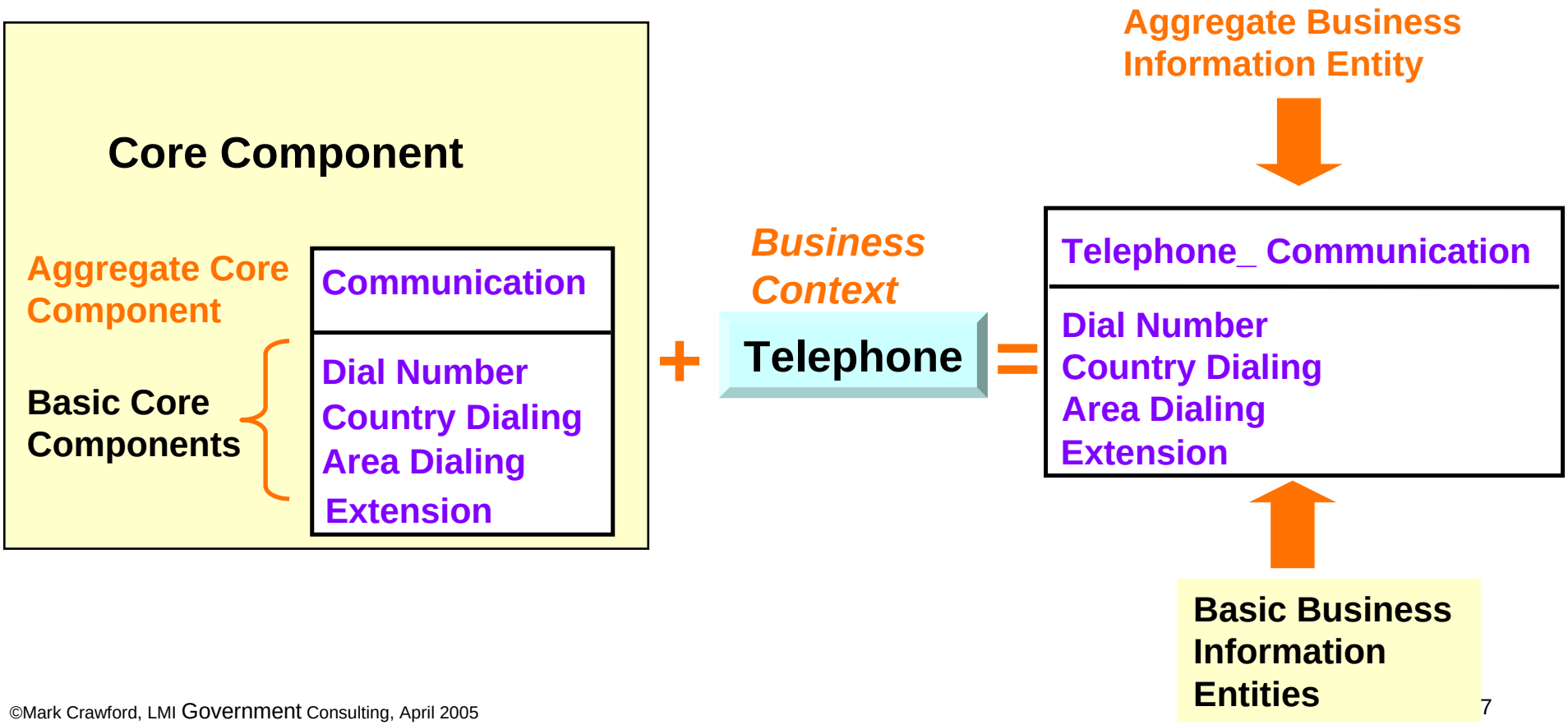
Basic BIE (BBIE)

- Property of an Aggregate Business Information Entity
- Based on a Basic Core Component (BCC)



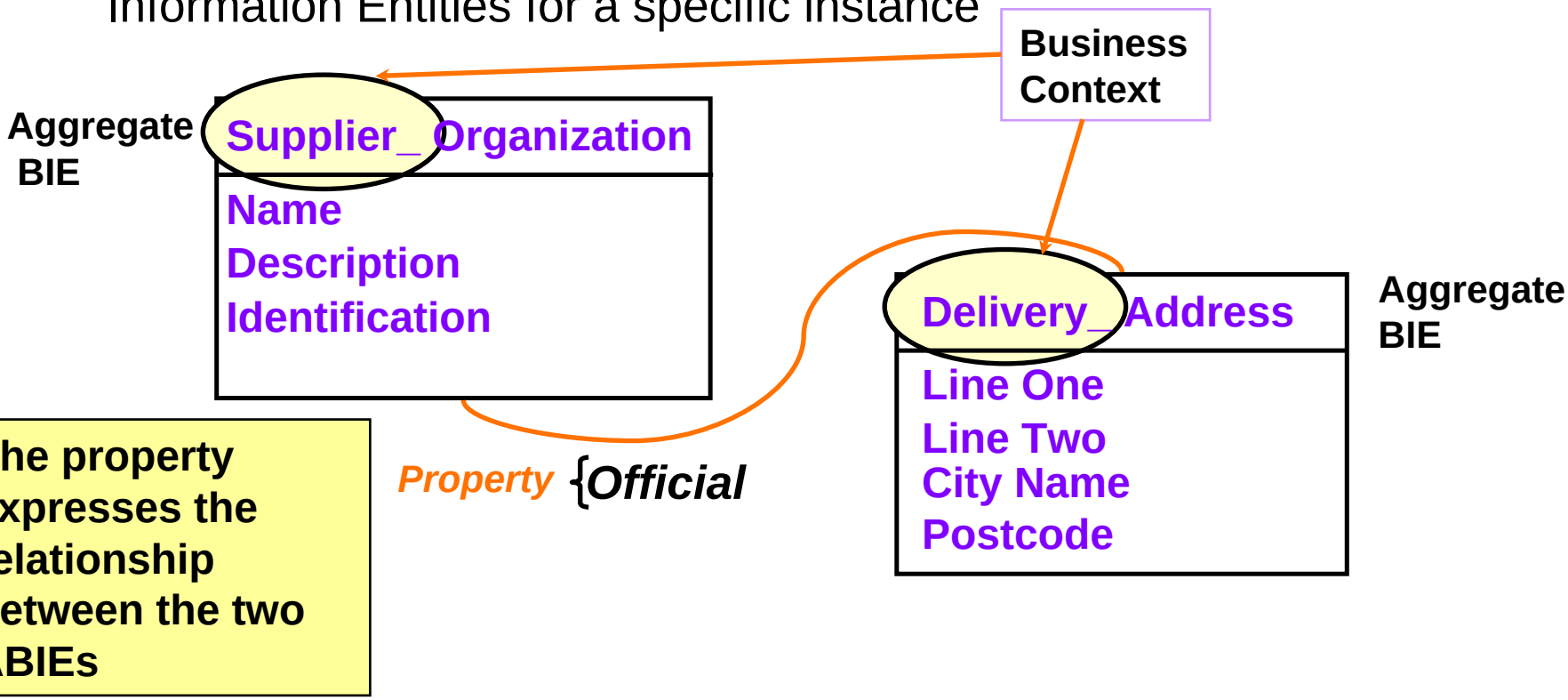
Aggregate BIE (ABIE)

- Representation of a Object Class
- An Aggregate Business Information Entity is a re-use of an Aggregate Core Component (ACC) in a specified Business context



Association BIE

- An ASBIE is a Business Information Entity naming mechanism for expressing the relationship between two Aggregate Business Information Entities for a specific instance

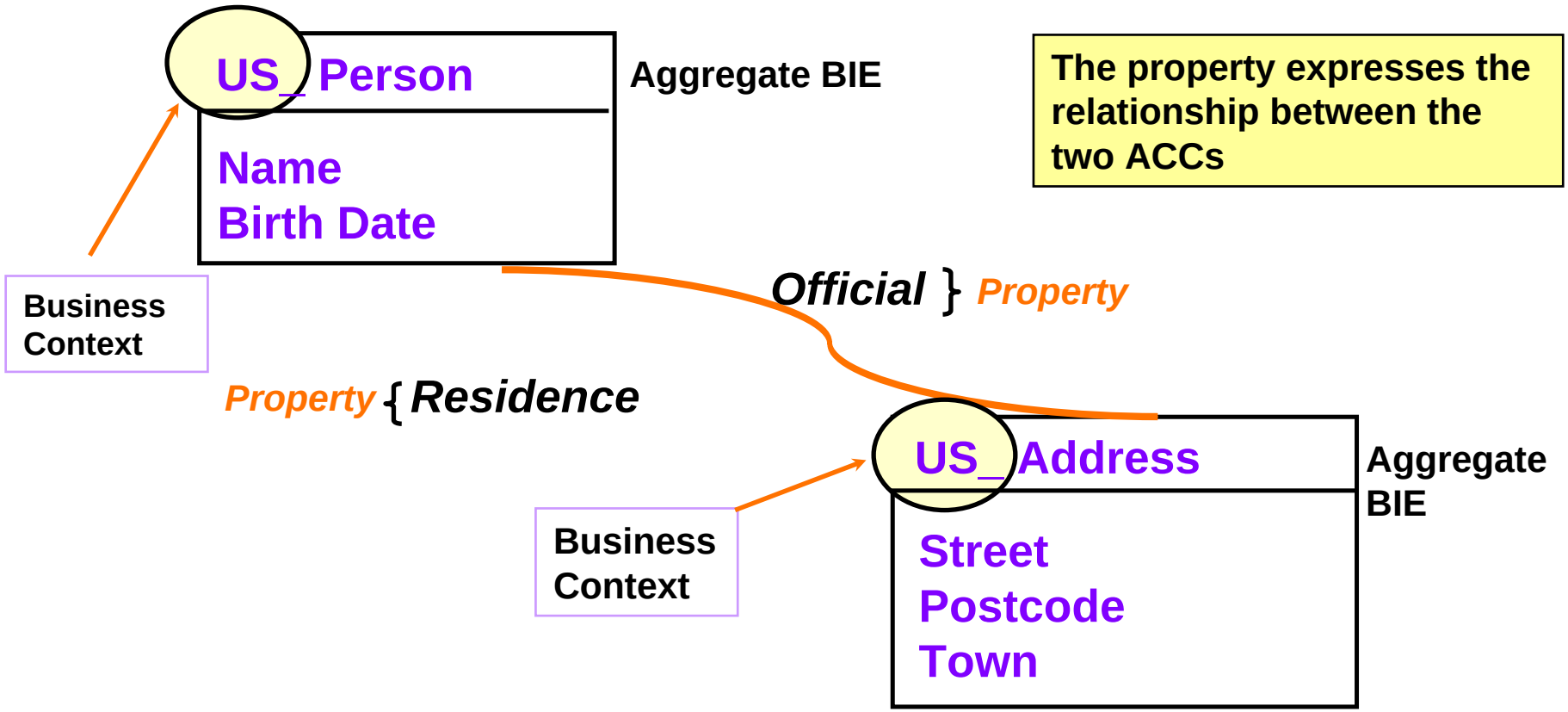


The property expresses the relationship between the two ABIEs

This supplier organization has an official delivery address Official is the property of the association

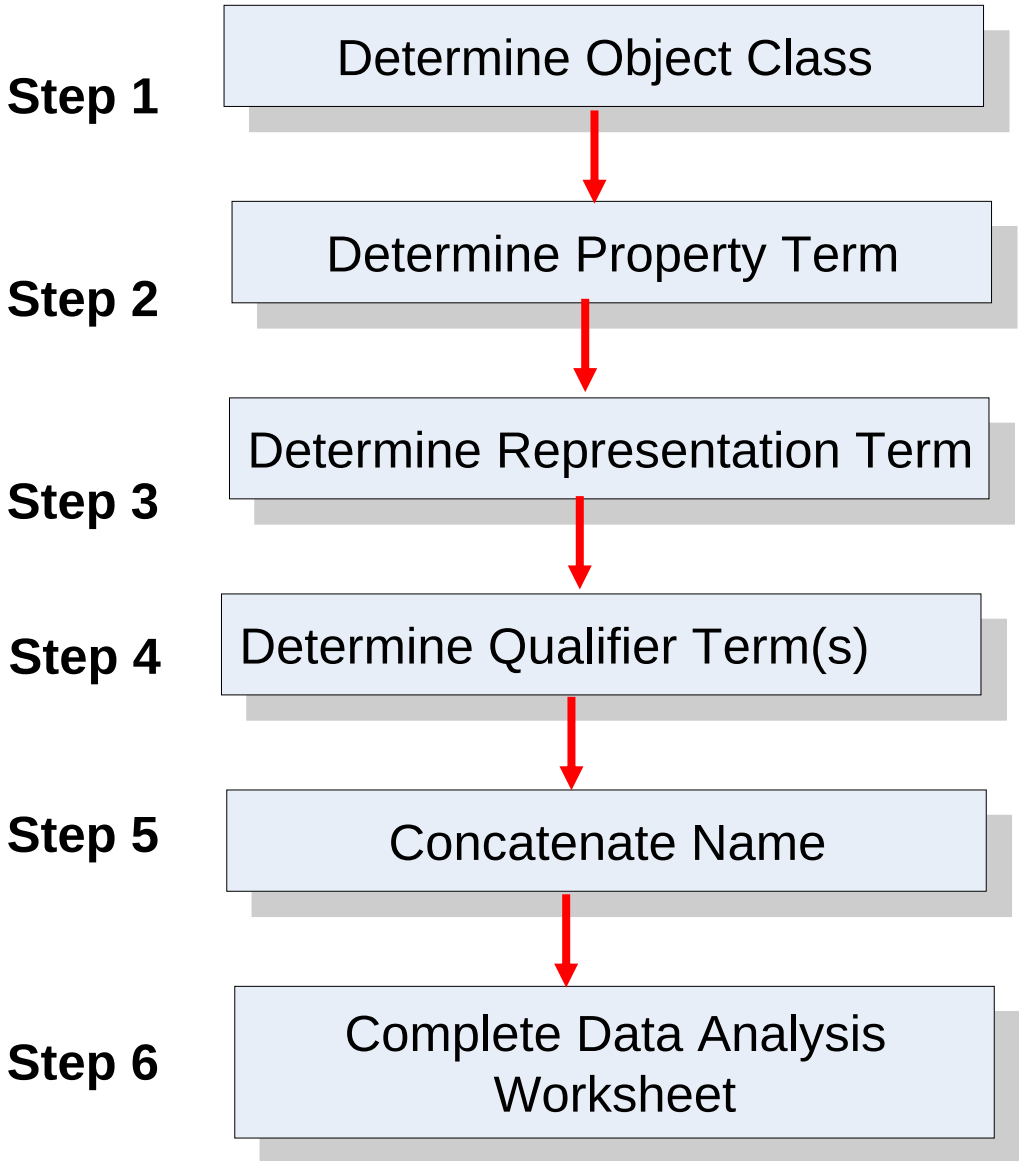
©Mark Crawford, LMI Government Consulting, April 2005

Association BIE Example



Business Information Entities
- **US_ Person. Residence. US_ Address (ASBIE)**
- **US_ Person. Official. US_ Address (ASBIE)**

Creating a Business Information Entity



©Mark Crawford, LMI Government Consulting, April 2005

Creating a Business Information Entity

LMI

Step 1: Determine Object Class

- Identify the logical grouping of elements
- Determine if this grouping has a singular business characteristic of an existing Aggregate Core Component in a specific Business Context
- Review controlled vocabulary of object class terms

Creating a Business Information Entity

LMI

Step 2: Determine Property Term

- Determine if the property distinguishes or describes the object class
- Determine if this property is a unique characteristic of the object class
- Review controlled vocabulary of property terms

Creating a Business Information Entity

LMI

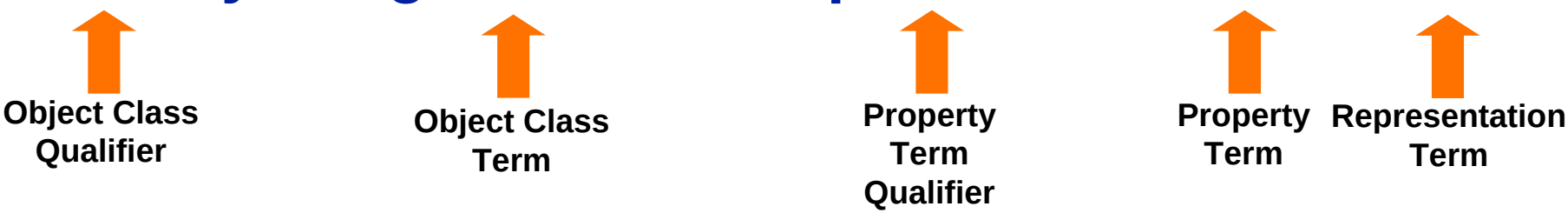
Step 3: Determine Representation Term

- Determine the nature of the atomic data type that reflects the use of this construct
- Using the atomic data type, Identify the appropriate permissible representation term from Table 8-3

Step 4: Determine Qualifiers

- Identify the business context for the component
- [B27] Qualifier Terms shall precede the associated Object Class Term or Property Term.

Inventory_ Organization. Department_ Name. Text



©Mark Crawford, LMI Government Consulting, April 2005

Creating a Business Information Entity

LMI

Step 5: Concatenate Name

- Comply with ISO 15000-5 Naming Rules

ISO 15000-5 ABIE DEN Rules

- [B30] The Dictionary Entry Name of an Aggregate Business Information Entity shall consist of the name of the Object Class of its associated Aggregate Core Component and possibly additional Qualifier Term(s) to represent its specific Business Context, followed by a dot, a space character, and the term Details

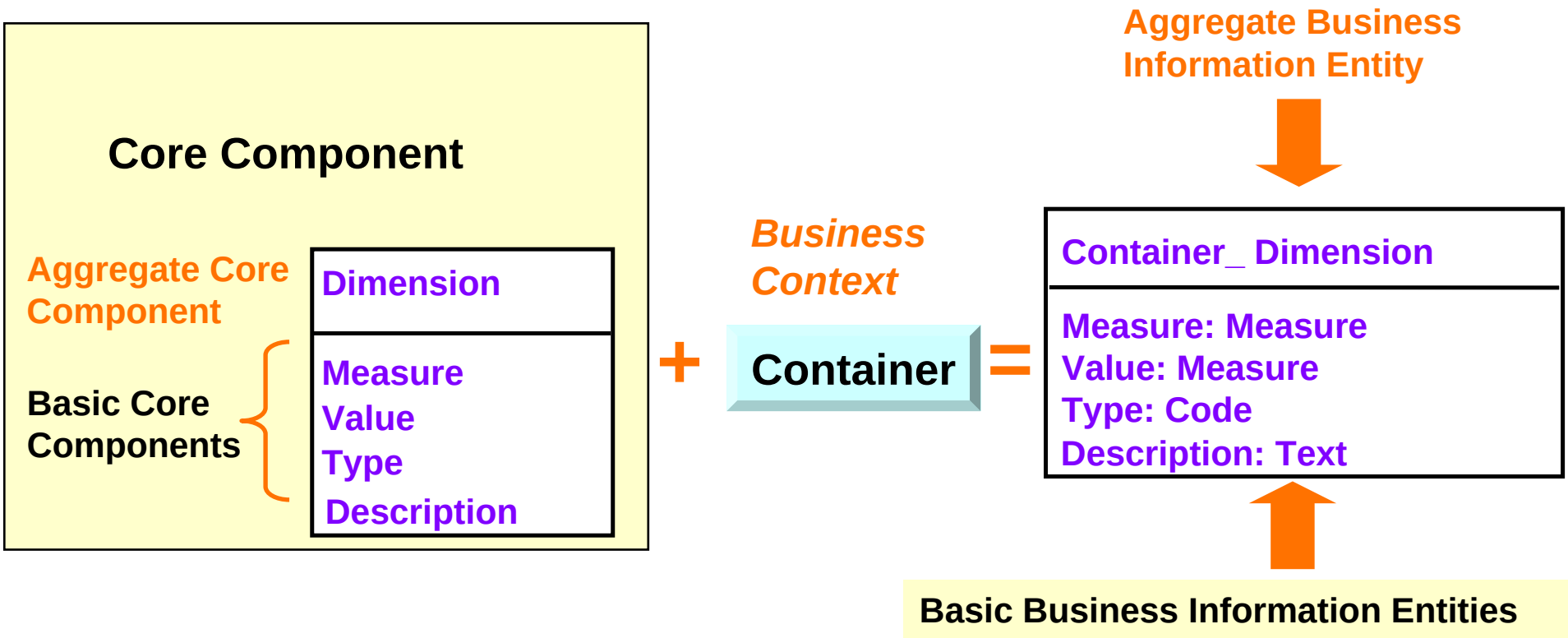
Supplier_ Organization. Details

Object Class Qualifier

Object Class

“Details”

Aggregate BIE Example



Container_ Dimension. Details (ABIE)

Qualifier Term

Object Class Term

Property Term

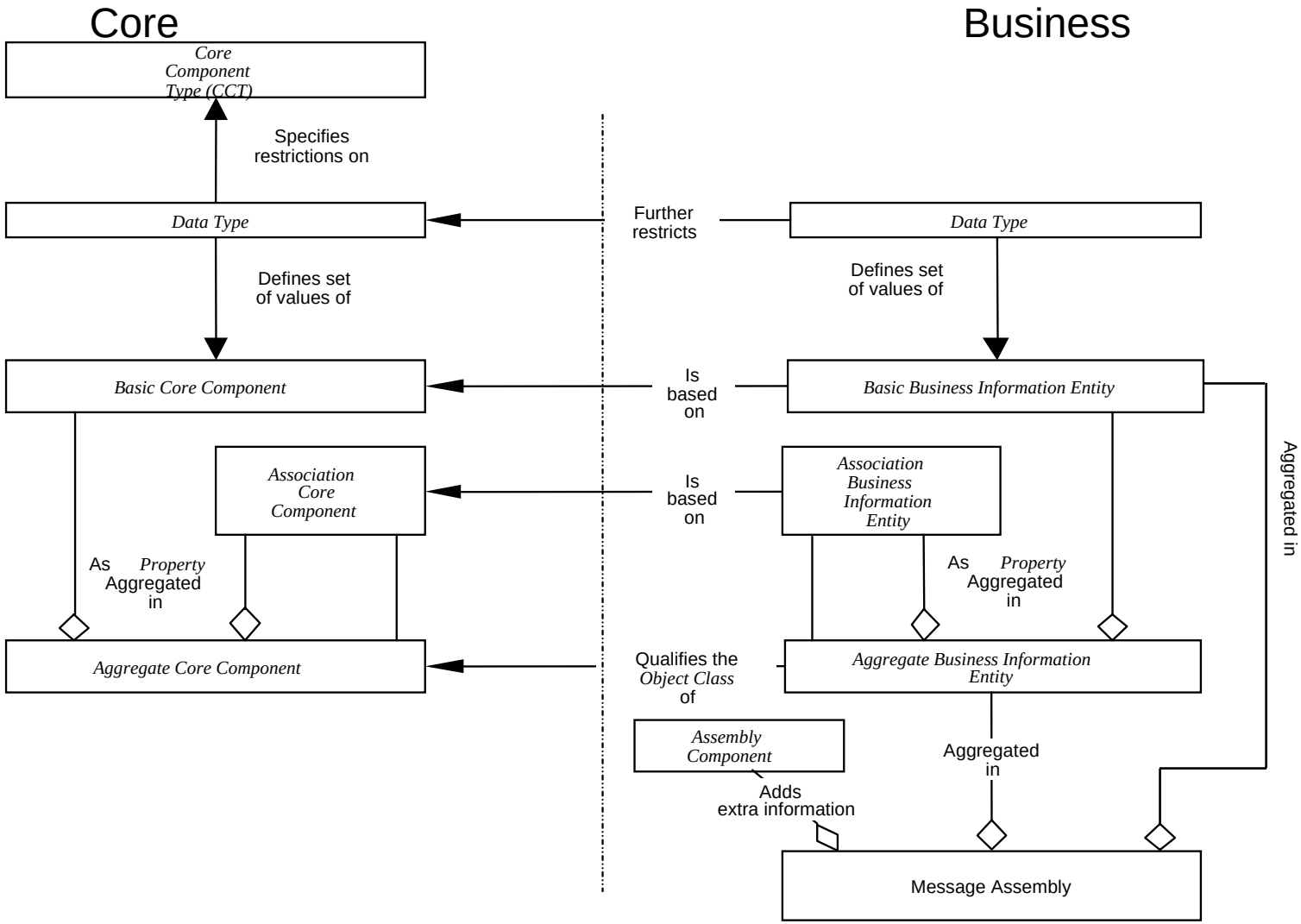
Data Analysis Worksheet

LMI

Step 6: Complete Data Analysis worksheet

Original Source Date Element Name	Definition	Dictionary Entry Name	ABIE/BBIE/ASBIE	Object Class Qualifier	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type	Associated Object Class Qualifier
	The information relevant to a person or organization that acts as a point of contact with	Supplier_ Contact. Details	ABIE	Supplier	Contact		Details		NA	
	The position or designation of this contact person within an organization such as Director, Software Engineer, Purchasing Manager.	Supplier_ Contact. Job Title. Text	BBIE	Supplier	Contact		Job Title	Text	TextType	
	The textual description of any general or specific responsibilities related to this contact.	Supplier_ Contact. Responsibility. Text	BBIE	Supplier	Contact		Responsibility	Text	TextType	

Tying It All Together



Core Component Library

A Reminder

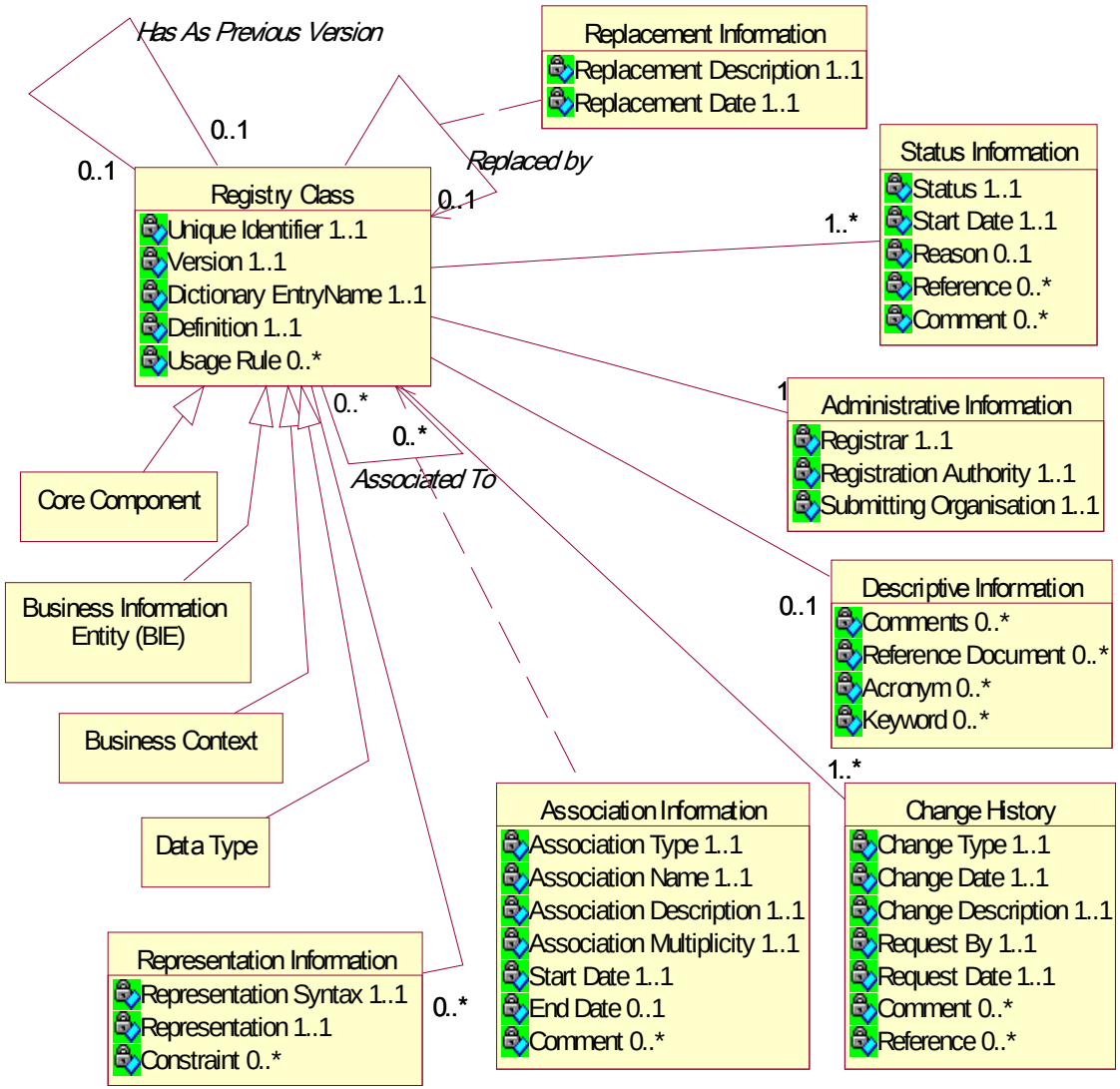
- The term Core Component is used as a generic term that encompasses Basic Core Components, Association Core Components, Aggregate Core Components, and their associated Core Component Types
- The term Business Information Entity is used as a generic term encompassing Basic Business Information Entities, Association Business Information Entities, and Aggregate Business Information Entities

Technical Details – Core Component Storage

LMI

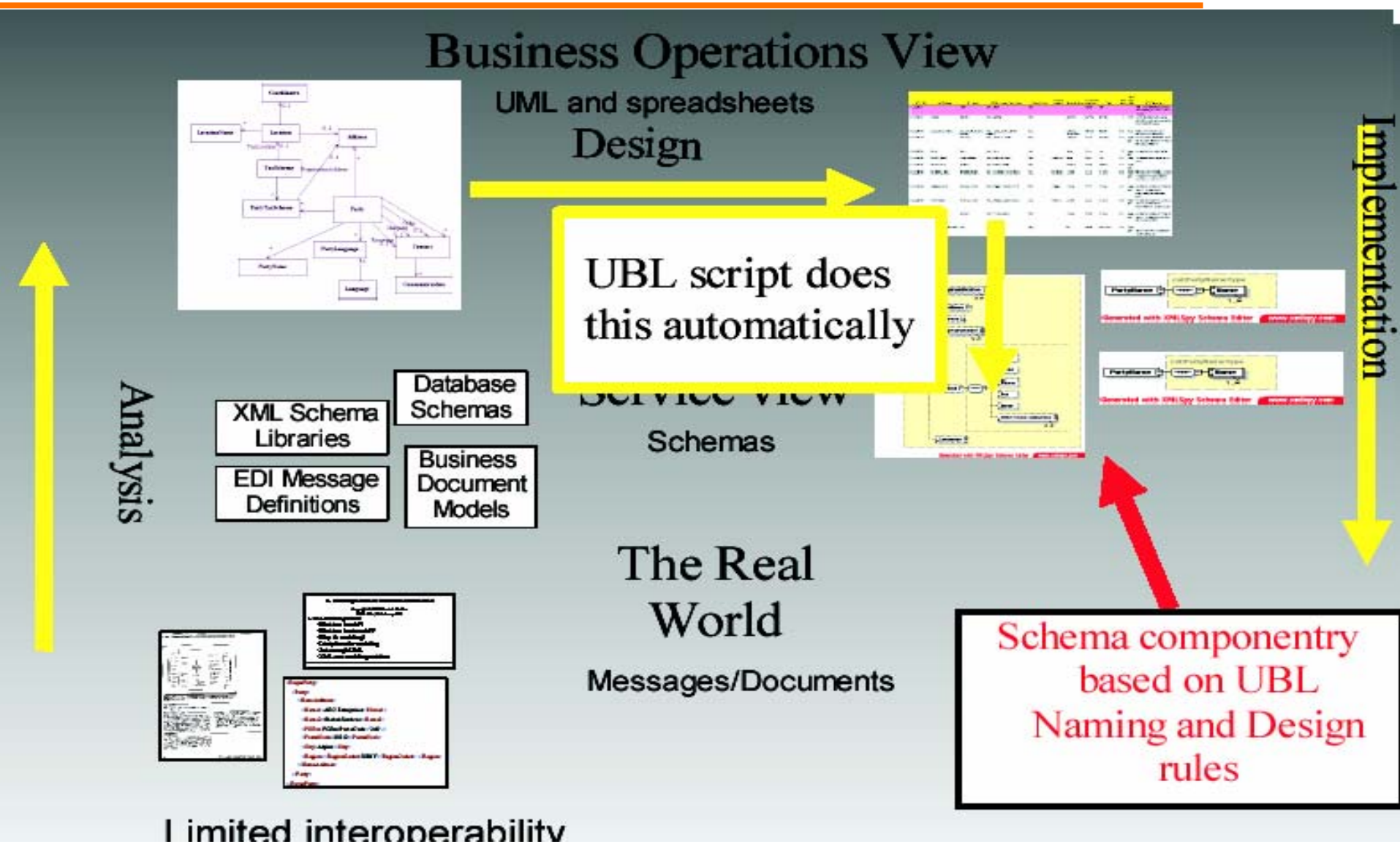
- Section 7 fully describes storage requirements for all Core Component and Business Information Entity Constructs
- The rules consist of
  - Storing Core Components
  - Storing Data Types
  - Storing Context
  - Storing Business Information Entities
  - Core Component Storage Metadata

The Storage Metamodel –Metadata



Creating the Schemas

LMI



A Schema Snippet - ItemType


```
<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:Representation
Term>
<ccts:BusinessTerm>article,product,goodsitem</ccts:Bu
sinessTerm>
</ccts:Component>
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element ref="Description" minOccurs="0">
...
...
...
```

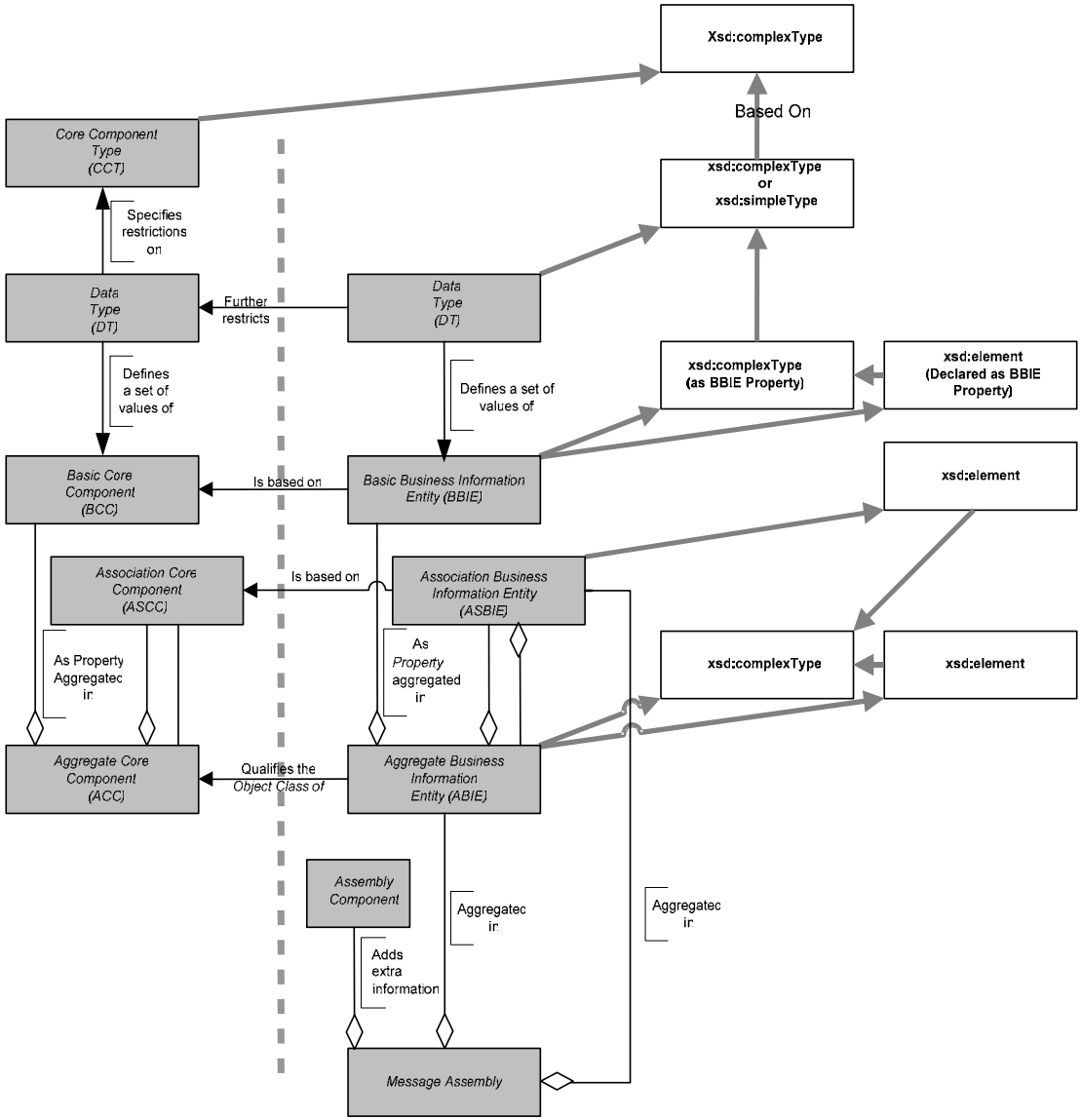
©Mark Crawford, LMI Government Consulting, April 2005

The Standards Used

LMI

- STA1 - All UBL schema design rules MUST be based on the W3C XML Schema Recommendations: XML Schema Part 1: Structures and XML Schema Part 2: Datatypes.
- STA2 - All UBL schema and messages MUST be based on the W3C suite of technical specifications holding recommendation status.

From ISO 15000-5 to Schema



Schema Structure

[GXS1] UBL Schema MUST conform to the following physical layout as applicable:

XML Declaration

<!-- ===== Copyright Notice ===== -->

<!-- ===== xsd:schema Element With Namespaces Declarations ===== -->

xsd:schema element to include version attribute and namespace declarations in the following order:

xmlns:xsd

Target namespace

Default namespace

CommonAggregateComponents

CommonBasicComponents

CoreComponentTypes

Unspecialized Datatypes

Specialized Datatypes

Identifier Schemes

Code Lists

Attribute Declarations – elementFormDefault=”qualified” attributeFormDefault=”unqualified”

<!-- ===== Imports ===== -->

CommonAggregateComponents schema module

CommonBasicComponents schema module

Unspecialized Types schema module

Specialized Types schema module

<!-- ===== Global Attributes ===== -->

Global Attributes and Attribute Groups

<!-- ===== Root Element ===== -->

Root Element

[ELD1] Each UBL:DocumentSchema MUST identify one and only one global element declaration that defines the document ccts:Aggregate BusinessInformationEntity being conveyed in the Schema expression. That global element MUST include an xsd:annotation child element which MUST further contain an xsd:documentation child element that declares *"This element MUST be conveyed as the root element in any instance document based on this Schema expression."*

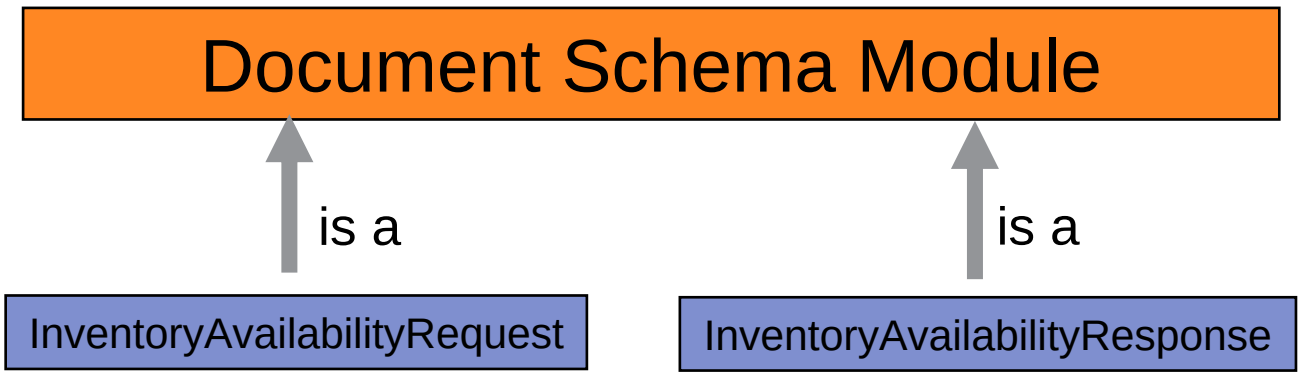
Example:

```
<xsd:element name="Order" type="OrderType">
  <xsd:annotation>
    <xsd:documentation>This element MUST be conveyed as the root element in
        any instance document based on this Schema
        expression</xsd:documentation>
  </xsd:annotation>
</xsd:element>
```

Document Schema

[Definition] Document schema –

The overarching schema within a specific namespace that conveys the business document functionality of that namespace. The document schema declares a target namespace and is likely to pull in by including internal schema modules or importing external schema modules. Each namespace will have one, and only one, document schema.



- Document does not denote / connote a ‘narrative’ document
- These XML ‘document’ Schemas define XML transactions for exchange between app servers

©Mark Crawford, LMI Government Consulting, April 2005

Element Declarations

LMI

[ELD2] All element declarations MUST be global with the exception of ID and Code which MUST be local.

- Much discussion on this issue
- Ultimate deciders were:
  - desire to manage by both types and elements
  - XPath limitations

Late Breaking News - UBL 2.0 will be all Global

Namespace Scheme

[NMS1] Every UBL-defined or -used schema module, except internal schema modules, MUST have a namespace declared using the xsd:targetNamespace attribute.

[NMS2] Every UBL-defined or -used schema set version MUST have its own unique namespace.

[Definition] Schema Set –

A collection of schema instances that together comprise the names in a specific UBL namespace.

[NMS3] UBL namespaces MUST only contain UBL developed schema modules.

Namespace URIs

LMI

- UBL has chosen URNs vice URLs as the Schema Location URI.
  - Primary differentiator is required run-time support and the need for persistence
  - Drawback is limit on URN resolvability
- RFC 2396 guides URI syntax
- RFC 3121 guides OASIS URN Namespace schemes

[NMS4] The namespace names for UBL Schemas holding committee draft status MUST be of the form:

urn:oasis:names:tc:ubl:schema:<subtype>:<document-id>

[NMS5] The namespace names for UBL Schemas holding OASIS Standard status MUST be of the form:

urn:oasis:names:specification:ubl:schema:<subtype>:<document-id>

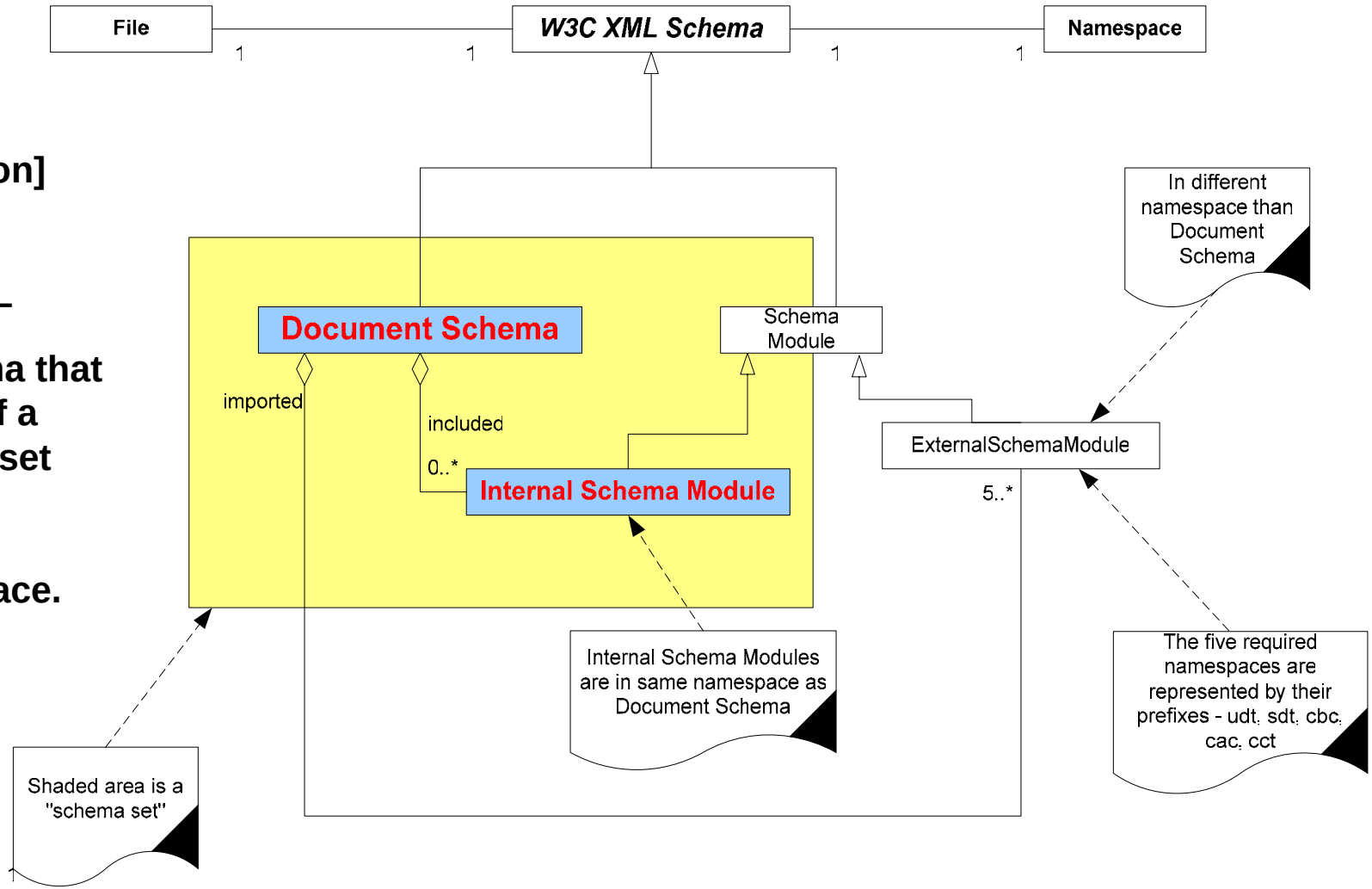
Versioning

- UBL has decided to include versioning information as part of the document-id component of the namespace
- The version information is divided into major and minor fields
- The minor field has an optional revision extension
- For example
  - urn:oasis:names:specification:ubl:schema:xsd:Order-1.0
- A host of rules related to standardizing this

Modularity Concept

[Definition] Internal schema module –

A schema that is part of a schema set within a specific namespace.

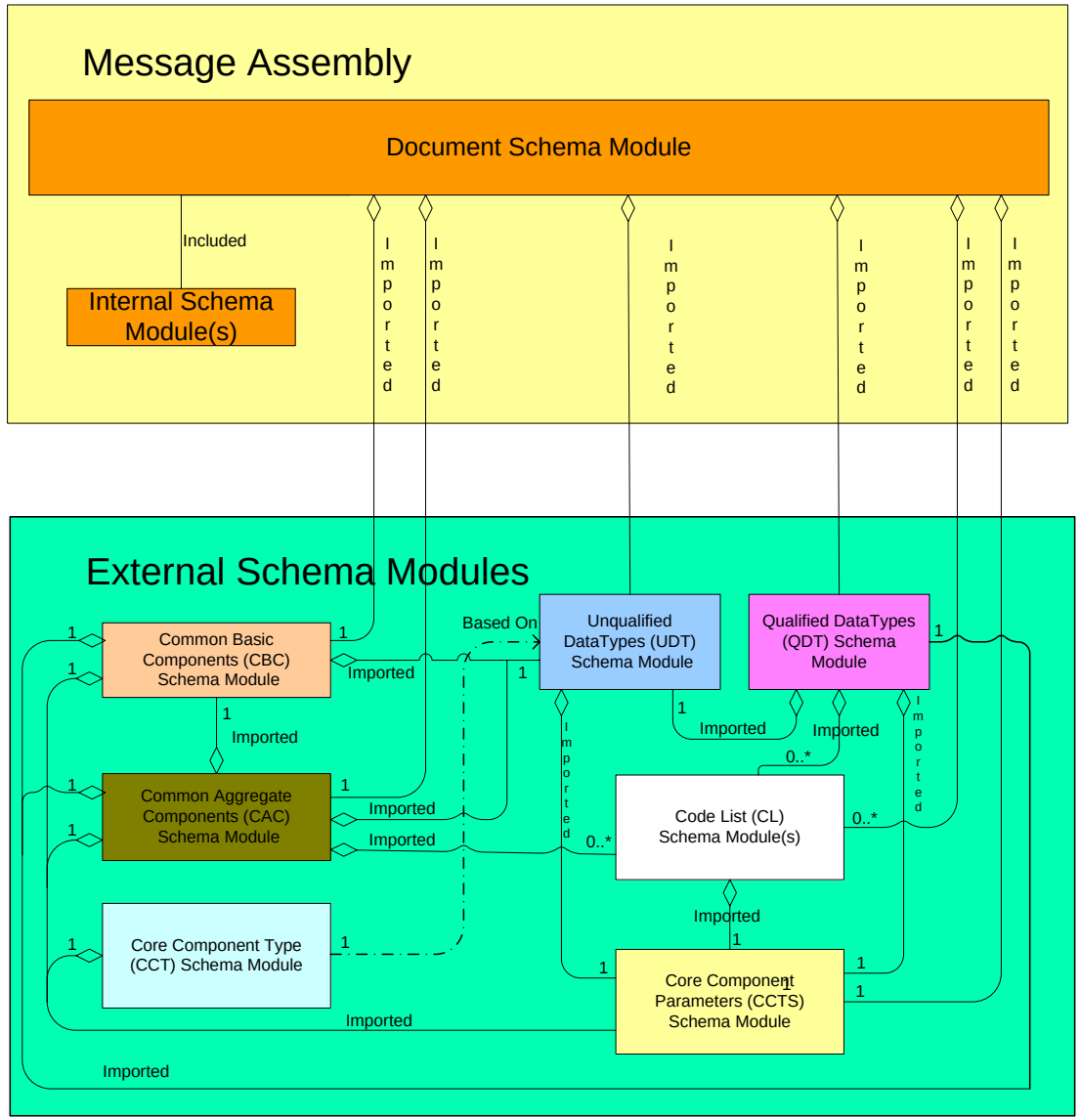


udt = Unspecialized Datatype, sdt = Specialized Datatype, cbc = Common Basic Components, cac = Common Aggregate Components, cct = Core Component Type

Modularity Architecture

[SSM6]

All UBL internal schema modules MUST be in the same namespace as their corresponding document schema.



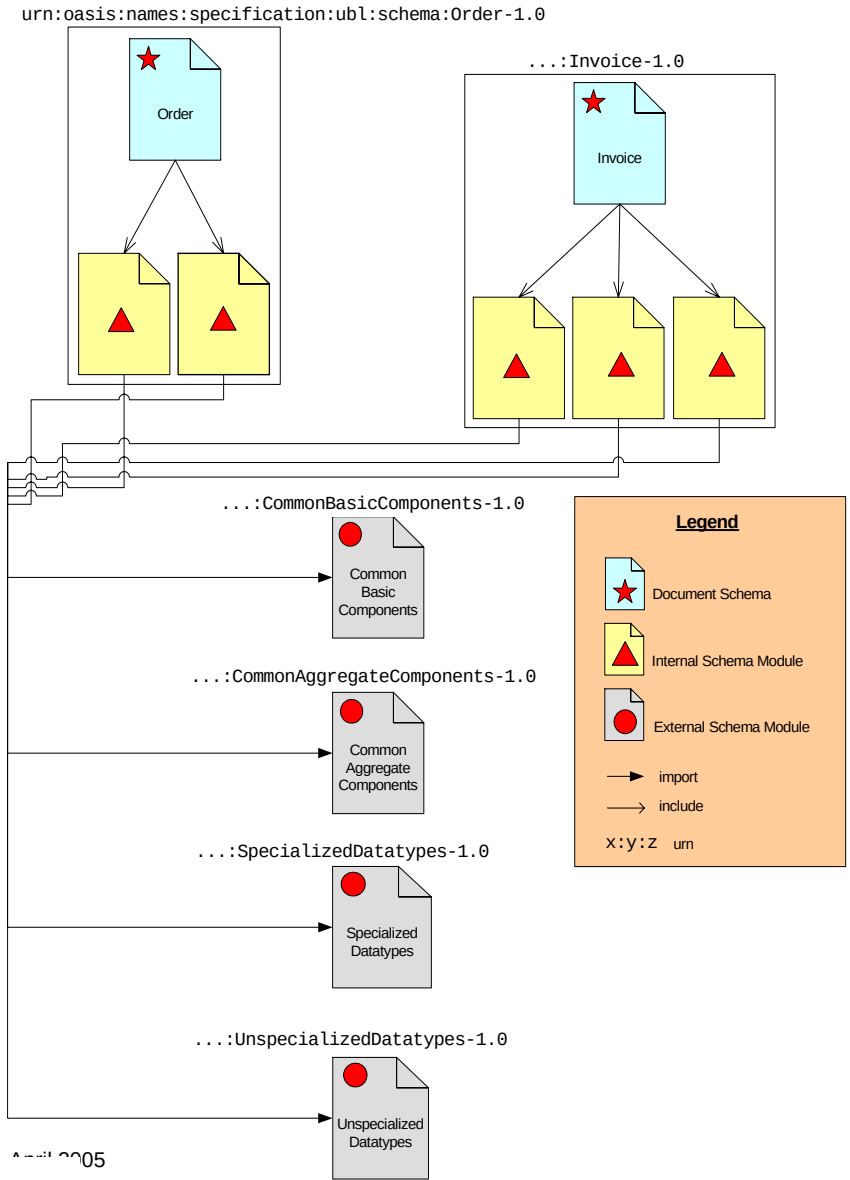
Limitations on Import

LMI

[SSM2] A document schema in one UBL namespace that is dependent upon type definitions or element declarations defined in another namespace MUST only import the document schema from that namespace.

[SSM3] A UBL document schema in one UBL namespace that is dependant upon type definitions or element declarations defined in another namespace MUST NOT import internal schema modules from that namespace.

Modularity In Action



Schema Modularity

LMI

[SSM1]	UBL Schema expressions MAY be split into multiple schema modules.
[SSM2]	A document schema in one UBL namespace that is dependent upon type definitions or element declarations defined in another namespace MUST only import the document schema from that namespace.
[SSM3]	A UBL document schema in one UBL namespace that is dependant upon type definitions or element declarations defined in another namespace MUST NOT import internal schema modules from that namespace.
[SSM4]	Imported schema modules MUST be fully conformant with UBL naming and design rules.
[SSM5]	UBL schema modules MUST either be treated as external schema modules or as internal schema modules of the document schema.
[SSM6]	All UBL internal schema modules MUST be in the same namespace as their corresponding document schema.
[SSM7]	Each UBL internal schema module MUST be named {ParentSchemaModuleName}{InternalSchemaModuleFunction}{schema module}
[SSM8]	A UBL schema module MAY be created for reusable components.
[SSM9]	A schema module defining all ubl:CommonAggregateComponents MUST be created.

Schema Modularity

LMI

[SSM10]	The ubl:CommonAggregateComponents schema module MUST be named "ubl:CommonAggregateComponents Schema Module"
[SSM11]	A schema module defining all ubl:CommonBasicComponents MUST be created.
[SSM12]	The ubl:CommonBasicComponents schema module MUST be named "ubl:CommonBasicComponents Schema Module"
[SSM13]	A schema module defining all ccts:CoreComponentTypes MUST be created.
[SSM14]	The ccts:CoreComponentType schema module MUST be named "ccts:CoreComponentType Schema Module"
[SSM15]	The xsd:facet feature MUST not be used in the ccts:CoreComponentType schema module.
[SSM16]	A schema module defining all ccts:UnspecialisedDatatypes MUST be created.
[SSM17]	The ccts:UnspecialisedDatatype schema module MUST be named "ccts:UnspecialisedDatatype Schema Module"

General Naming

- Top-level element:
  - An element that encloses a whole UBL business message. Note that UBL business messages might be carried by messaging transport protocols that themselves have higher-level XML structure. Thus, a UBL top-level element is not necessarily the root element of the XML document that carries it.
- Lower-level element:
  - An element that appears inside a UBL business message. Lower-level elements consist of intermediate and leaf level.
- Intermediate element:
  - An element not at the top level that is of a complex type, only containing other elements and attributes.
- Leaf element:
  - An element containing only character data (though it may also have attributes). Note that, because of the XSD mechanisms involved, a leaf element that has attributes must be declared as having a complex type, but a leaf element with no attributes may be declared with either a simple type or a complex type.
- Common attribute:
  - An attribute that has identical meaning on the multiple elements on which it appears. A common attribute might or might not correspond to an XSD global attribute.

©Mark Crawford, LMI Government Consulting, April 2005

General Naming Rules

LMI

[GNR1]	UBL XML element, attribute and type names MUST be in the English language, using the primary English spellings provided in the Oxford English Dictionary.
[GNR2]	UBL XML element, attribute and type names MUST be consistently derived from CCTS conformant dictionary entry names.
[GNR3]	UBL XML element, attribute and type names constructed from ccts:DictionaryEntryNames MUST NOT include periods, spaces, other separators, or characters not allowed by W3C XML 1.0 for XML names.
[GNR4]	UBL XML element, attribute, and simple and complex type names MUST NOT use acronyms, abbreviations, or other word truncations, except those in the list of exceptions published in Appendix B.
[GNR5]	Acronyms and abbreviations MUST only be added to the UBL approved acronym and abbreviation list after careful consideration for maximum understanding and reuse.
[GNR6]	The acronyms and abbreviations listed in Appendix B MUST always be used.
[GNR7]	UBL XML element, attribute and type names MUST be in singular form unless the concept itself is plural.
[GNR8]	The UpperCamelCase (UCC) convention MUST be used for naming elements and types.
[GNR9]	The lowerCamelCase (LCC) convention MUST be used for naming attributes.

ABIE Element and complexType Naming -

[CTN1]	A UBL xsd:complexType name based on an ccts:AggregateBusinessInformationEntity MUST be the ccts:DictionaryEntryName with the separators removed and with the "Details" suffix replaced with "Type".
[ELN1]	A UBL global element name based on a ccts:ABIE MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.

XML Schema Model

CCTS Model

XML Tag Name	XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE
InventoryAvailabilityRequest	InventoryAvailabilityRequestType	InventoryAvailability_ Request. Details	ABIE

BBIE Property Element and complexType Naming

[ELN2]	A UBL global element name based on an unqualified ccts:BBIEProperty MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.
[CTN2]	A UBL xsd:complexType name based on a ccts:BasicBusinessInformationEntityProperty MUST be the ccts:DictionaryEntryName shared property term and its qualifiers and the representation term of the shared ccts:BasicBusinessInformationEntity, with the separators removed and with the "Type" suffix appended after the representation term.

XML Schema Model **CCTS Model**

XML Tag Name	XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE
Name	NameType	Supplier_ Organization. Name. Text	**BBIE**

BBIE Property Element and complexType Naming

[ELN2]	A UBL global element name based on an unqualified ccts:BBIEProperty MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.

[ELN4] A UBL global element name based on a qualified ccts:BBIEProperty MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the qualifier prefixed and with the word "Type" removed.

XML Schema Model **CCTS Model**

XML Tag Name	XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE
Name	NameType	Supplier_ Organization. Name. Text	**BBIE**
DepartmentName	DepartmentNameType	Supplier_ Organization. Department_ Name. Text	**BBIE**
DepartmentID	DepartmentIDType	Supplier_ Organization. Department_ Identification. Identifier	**BBIE**

ASBIE Element Naming

[ELN3]	A UBL global element name based on a qualified ccts:ASBIE MUST be the ccts:ASBIE dictionary entry name property term and its qualifiers; and the object class term and qualifiers of its associated ccts:ABIE. All ccts:DictionaryEntryName separators MUST be removed. Redundant words in the ccts:ASBIE property term or its qualifiers and the associated ccts:ABIE object class term or its qualifiers MUST be dropped.

XML Schema Model | **CCTS Model**

XML Tag Name	XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ ASBIE /BBIE
OfficialContactMailingAddress	N/A	Supplier_ Organization. Official_ Contact. Mailing_ Address	**ASBIE**

Note: No CT for ASBIE

ASBIE Element Naming

[ELN3]	A UBL global element name based on a qualified ccts:ASBIE MUST be the ccts:ASBIE dictionary entry name property term and its qualifiers; and the object class term and qualifiers of its associated ccts:ABIE. All ccts:DictionaryEntryName separators MUST be removed. Redundant words in the ccts:ASBIE property term or its qualifiers and the associated ccts:ABIE object class term or its qualifiers MUST be dropped.

```
<xsd:complexType name="ForecastUpdateType">
        <xsd:annotation>
                <xsd:documentation>
                <ccts:CategoryCode>ABIE</ccts:CategoryCode>
                <ccts:DictionaryEntryName> Forecast_ Update. Details</ccts:DictionaryEntryName>
                <ccts:Definition>ForecastUpdate</ccts:Definition>
                <ccts:ObjectClass>Update</ccts:ObjectClass>
                <ccts:RepresentationTerm>TBD</ccts:RepresentationTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
                <xsd:element ref="cac:ForecastSupplierOrganization"/>
        </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

Note: Truncation rules apply to ASBIEs

Unspecialized Datatypes ComplexType Naming

LMI

[CTN3] A UBL xsd:complexType for a cct:UnspecializedDatatype used in the UBL model MUST have the name of the corresponding ccts:CoreComponentType, with the separators removed and with the “Type” suffix appended.

Example:

```
<!-- ===== Primary Representation Term: AmountType ===== -->
<xsd:complexType name="AmountType">
        ...
</xsd:complexType>
```

Unspecialized Datatypes ComplexType Naming

[CTN4] A UBL xsd:complexType for a cct:UnspecializedDatatype based on a ccts:SecondaryRepresentationTerm used in the UBL model MUST have the name of the corresponding ccts:SecondaryRepresentation
Term, with the separators removed and with the “Type” suffix appended.

Example:

```
<!-- ===== Secondary Representation Term: GraphicType ===== -->
<xsd:complexType name="GraphicType">
                    ...
</xsd:complexType>
```

CCT ComplexType Naming

[CTN5] A UBL xsd:complexType name based on a ccts:CoreComponentType MUST be the Dictionary entry name of the ccts:CoreComponentType, with the separators removed.

Example:

```
<!-- ===== CCT: QuantityType ===== -->
<xsd:complexType name="QuantityType">
        ...
</xsd:complexType>
```

Attribute Naming

LMI

[ATN1] Each CCT:SupplementaryComponent xsd:attribute "name" MUST be the Dictionary Entry Name object class, property term and representation term of the ccts:SupplementaryComponent with the separators removed.

Example:

ccts:SupplementaryComponent	ubl:attribute
Amount Currency.Identifier	amountCurrencyID
Amount Currency. Code List Version.Identifier	amountCurrencyCodeListVersionID
Measure Unit.Code	measureUnitCode

Type Definitions

[GTD1] All types MUST be named

Example:

```
<xsd:complexType name="QuantityType">
    ...
</xsd:complexType>
```

[GTD2] The xsd:anyType MUST NOT be used

CCT SimpleType Definition

[STD1] For every ccts:CCT whose supplementary components map directly onto the properties of a built-in xsd:Datatype, the ccts:CCT MUST be defined as a named xsd:simpleType in the ccts:CCT schema module.

Example:

```
<!-- ===== CCT: DateTimeType ===== -->
<xsd:simpleType name="DateTimeType">
        ...
        <xsd:restriction base="cct:DateTimeType"/>
</xsd:simpleType>
```

Complex Type Definitions

[CTD1] For every class identified in the UBL model, a named xsd:complexType MUST be named

XML Schema Model		CCTS Model	
XML Tag Name	**XML ComplexType Name**	**CCTS Dictionary Entry Name**	**ABIE/ASBIE/BBIE**
InventoryAvailabilityRequest	**InventoryAvailabilityRequestType**	**InventoryAvailability_ Request. Details**	**ABIE**

Complex Type Definition - ABIE

LMI

[CTD2] Every ccts:ABIE xsd:complexType definition content model MUST use the xsd:sequence element with appropriate global element references, or local element declarations in the case of ID and Code, to reflect each property of its class as defined in the corresponding UBL model.

```
<xsd:complexType name="InventoryItemType">
        <xsd:sequence>
                <xsd:element name="InventoryItemID" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="cbc:Quantity" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="cbc:Name" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
</xsd:complexType>
```

Complex Type Definition – BBIE Property

[CTD3] **Every ccts:BBIEProperty xsd:complexType definition content model MUST use the xsd:simpleContent element.**

[CTD4] **Every ccts:BBIEProperty xsd:complexType content model xsd:simpleContent element MUST consist of an xsd:extension element.**

```
<xsd:complexType name="DepartmentNameType">
    <xsd:simpleContent>
        <xsd:extension base="udt:TextType"/>
    </xsd:simpleContent>
</xsd:complexType>
```

[CTD5] **Every ccts:BBIEProperty xsd:complexType content model xsd:base attribute value MUST be the ccts:CCT of the unspecialized or specialized UBL Datatype as appropriate.**

Core Component type – Type Definitions

[CTD13] For every ccts:CCT whose supplementary components are not equivalent to the properties of a built-in xsd:Datatype, the ccts:CCT MUST be defined as a named xsd:complexType in the ccts:CCT schema module.

Example:

```
<xsd:complexType name="QuantityType">
    <xsd:simpleContent>
     <xsd:extension base="xsd:decimal">
       <xsd:attribute name="quantityUnitCode" type="xsd:normalizedString" use="optional"/>
       <xsd:attribute name="quantityUnitCodeListID" type="xsd:normalizedString"
  use="optional"/>
       <xsd:attribute name="quantityUnitCodeListAgencyID" type="xsd:normalizedString"
  use="optional"/>
       <xsd:attribute name="quantityUnitCodeListAgencyName" type="xsd:string"
  use="optional"/>
     </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

©Mark Crawford, LMI Government Consulting, April 2005

Handling Supplementary Components

[CTD16] Each CCT:SupplementaryComponent xsd:attribute “type” MUST define the specific xsd:Built-inDatatype or the user defined xsd:simpleType for the ccts:SupplementaryComponent of the ccts:CCT.

Example:

```
<xsd:attribute name="measureUnitCode"
               type="xsd:normalizedString" use="required"/>
```

Datatype complex and simpleType Definitions

- There is a direct one-to-one relationship between ccts:CoreComponentTypes and ccts:PrimaryRepresentationTerms
  - several ccts:SecondaryRepresentationTerms that are subsets of their parent ccts:PrimaryRepresentationTerm
- The total set of ccts:Representation Terms by their nature represent ccts:Datatypes
- For each ccts:PrimaryRepresentationTerm or ccts:SecondaryRepresentationTerm, a ccts:UnspecializedDatatype exists
- These ccts:UnspecializedDatatypes are expressed as complex or simple types that are of the type of its corresponding ccts:CoreComponentType.

[CTD6] For every Datatype used in the UBL model, a named xsd:complexType or xsd:simpleType MUST be defined.

Datatype complexType and simpleType Definitions

LMI

[CTD7] Every unspecialized Datatype must be based on a ccts:CCT represented in the CCT schema module, and must represent an approved primary or secondary representation term identified in the CCTS.

[CTD8] Each unspecialized Datatype xsd:complexType must be based on its corresponding CCT xsd:complexType.

[CTD9] Every unspecialized Datatype that represents a primary representation term whose corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.

[CTD10] Every unspecialized Datatype that represents a secondary representation term whose corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.

Element Declarations

[ELD3] For every class identified in the UBL model, a global element bound to the corresponding xsd:complexType MUST be declared.

Example:

For the BuyerParty. Details object class, a complex type/global element declaration pair is created through the declaration of a Party element that is of type BuyerPartyType.

```
<xsd:element name="BuyerParty" type="BuyerPartyType"/>
<xsd:complexType name="BuyerPartyType">
                    ...
</xsd:complexType>
```

ASBIE Element Declaration

[ELD4] When a ccts:ASBIE is **unqualified**, it is bound via reference to the global ccts:ABIE element to which it is associated. When an ccts:ASBIE is qualified, a new element MUST be declared and bound to the xsd:complexType of its associated ccts:AggregateBusinessInformationEntity.

```
<xsd:complexType name="InventoryAvailabilityRequestType">
        <xsd:annotation>
            <xsd:documentation>
                <ccts:CategoryCode>ABIE</ccts:CategoryCode>
                <ccts:DictionaryEntryName/>
                <ccts:Definition>This holds information specific for an inventory
availability request.</ccts:Definition>
                <ccts:ObjectClass/>
                <ccts:RepresentationTerm/>
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="cac:InventoryOrganization"/>
            <xsd:element ref="cac:InventoryItem" maxOccurs="unbounded"/>
            <xsd:element ref="cac:SupplierOrganization"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

ASBIE Element Declaration

[ELD4] When a ccts:ASBIE is unqualified, it is bound via reference to the global ccts:ABIE element to which it is associated. When an ccts:ASBIE is **qualified**, a new element MUST be declared and bound to the xsd:complexType of its associated ccts:AggregateBusinessInformationEntity.

```
<xsd:complexType name="ForecastUpdateType">
                    <xsd:annotation>
                        <xsd:documentation>
                            <ccts:CategoryCode>ABIE</ccts:CategoryCode>
                            <ccts:DictionaryEntryName>Forecast_ Update.
Details</ccts:DictionaryEntryName>
                            <ccts:Definition>ForecastUpdate</ccts:Definition>
                            <ccts:ObjectClass>Update</ccts:ObjectClass>
        <ccts:RepresentationTerm>TBD</ccts:RepresentationTerm>
                        </xsd:documentation>
                    </xsd:annotation>
                    <xsd:sequence>
                        <xsd:element ref="cac:ForecastSupplierOrganization"/>
                    <xsd:annotation>…….</xsd:annotation>
                    </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

ASBIE

BBIE Property Element Declarations

[ELD8] Global elements declared for Qualified BBIE Properties must be of the same type as its corresponding Unqualified BBIE Property. (i.e. Property Term + Representation Term.)

Example:

```
<xsd:element name="AdditionalStreetName"
        type="cbc:StreetNameType"/>
```

Attribute Declarations

[ATD1] User defined attributes SHOULD NOT be used. When used, user defined attributes MUST only convey CCT:SupplementaryComponent information.

Code Lists

[CDL1] All UBL Codes MUST be part of a UBL or externally maintained Code List.

[CDL2] The UBL Library SHOULD identify and use external standardized code lists rather than develop its own UBL-native code lists.

[CDL3] The UBL Library MAY design and use an internal code list where an existing external code list needs to be extended, or where no suitable external code list exists.

Documentation Rules

[GXS2] UBL MUST provide two normative schemas for each transaction. One schema shall be fully annotated. One schema shall be a run-time schema devoid of documentation.

Documentation Rules

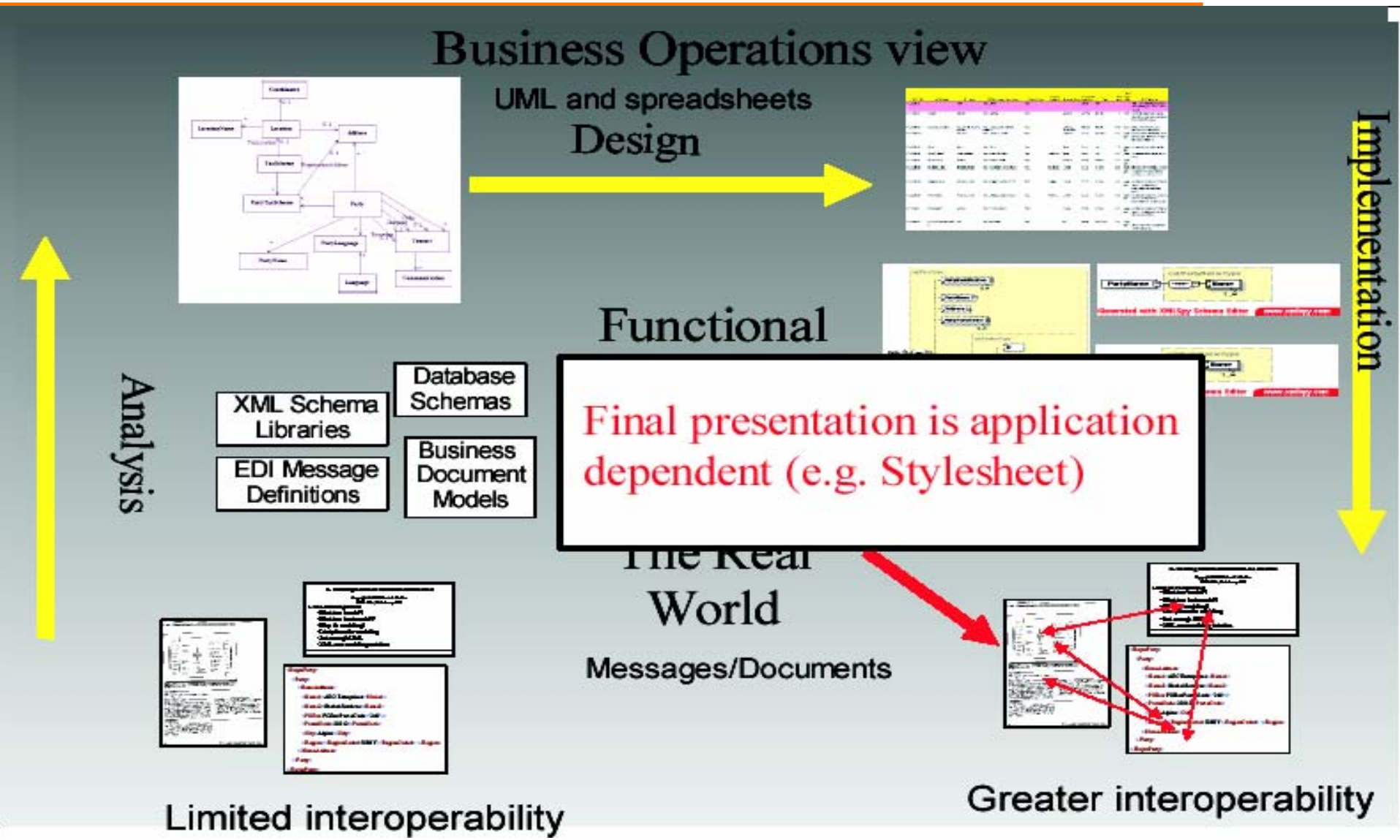
- Every CCTS construct must contain all mandatory CCTS Section 7 storage metadata
- Every element declaration and type definition in a UBL model must include all mandatory CCTS Section 7 Storage metadata
- Example:

[DOC1] Every data type definition MUST contain a structured set of annotations in the following sequence and pattern:

  - UniqueIdentifier (mandatory): The identifier that references a data type instance in a unique and unambiguous way.
  - CategoryCode (mandatory): The category to which the object belongs. For example, BBIE, ABIE, ASBIE.
  - DictionaryEntryName (mandatory): The official name of a data type.
  - Definition (mandatory): The semantic meaning of a data type.
  - Version (mandatory): An indication of the evolution over time of a data type instance.
  - QualifierObjectClass (optional): The qualifier for the object class.
  - Usage Rule (optional, repetitive): A constraint that describes specific conditions that are applicable to the data type.

©Mark Crawford, LMI Government Consulting, April 2005

Implementation Models



Formatting Specifications

LMI

- Formatting Specification in Detail
  - A formatting specification is a recipe for a stylesheet, but is not in and of itself a transformation script.
  - Writers of stylesheets, programs, or any other open and proprietary transformation technologies rely on formatting specifications for direction regarding content identification and layout.
- PDF Renderings of example instances

Example Formatting Specification

LMI

- **Despatch advice formatting specifications**
  - Three sample formatting specifications are offered for this document type:
    - Office-oriented despatch advice form
    - Joinery-oriented despatch advice form
    - United Nations Layout Key form 351: Despatch Advice

Sample UBL Document Instance

Invoice
Invoice Number: 9834562
Invoice Date: 02-14-03
Purchase Order No: 20031234-1
Sales Order Number: 154135798
Shipment Date: 02-14-03

To: Bills Microdevices
413 Spring St.
Elgin, Ill 60123

From: Joes Office Supply
32 W. Lakeshore Dr
Chicago, Ill 60022

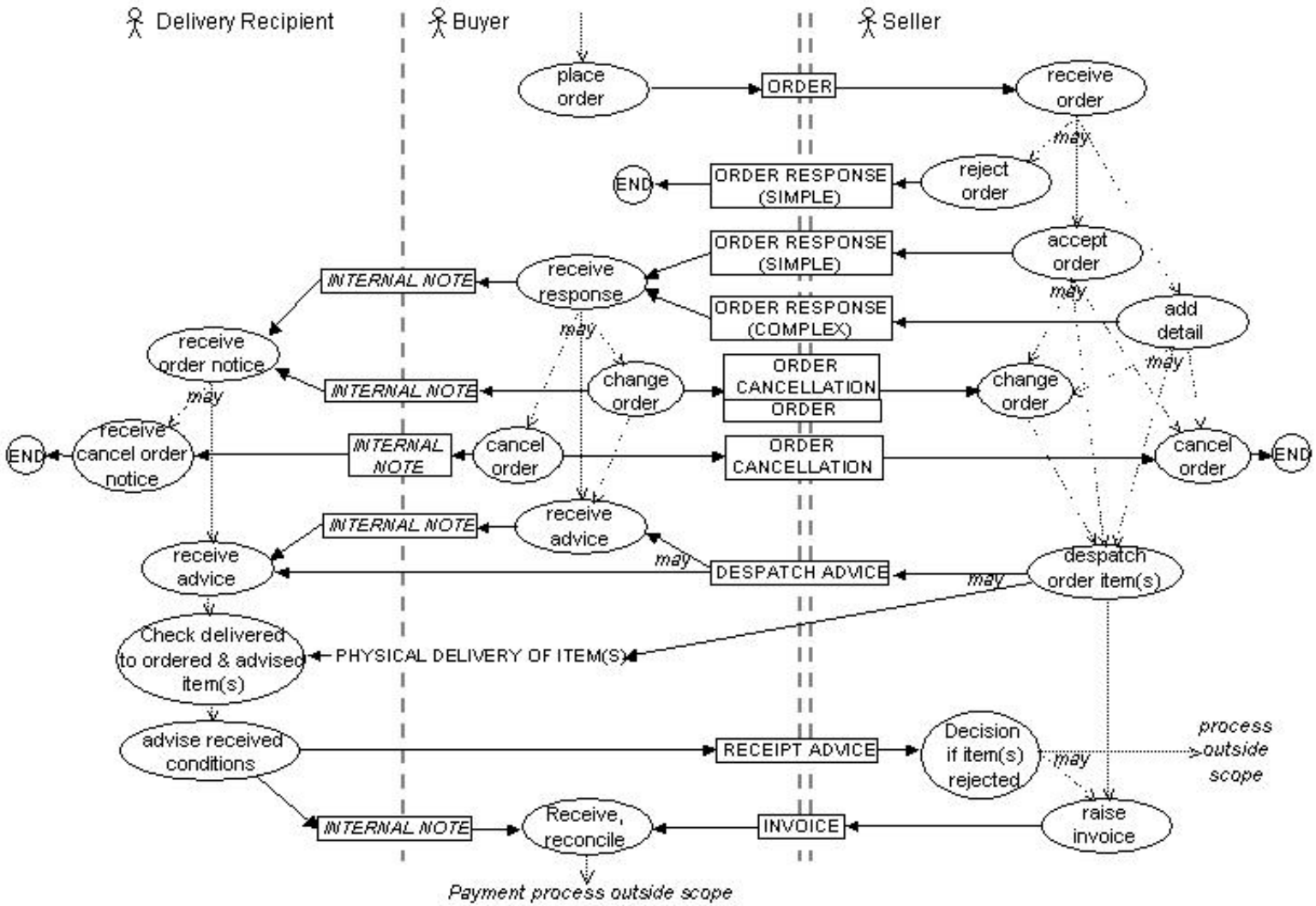
Billing Contact: Melanie Farber (312) 865-2199

Shipped to: 413 N Spring St.
Elgin, Ill 60123

Line Num	*Part Number*	*Description*	*Qty*	*Unit Price*	*Extended Amount*
1	32145-12	Pencils, box #2 red	5	$2.50	$12.50
2	78-697-24	Xerox Paper- case	12	$30.00	$360.00
3	091356-3	Pens, box, blue finepoint	10	$5.00	$50.00
4	543-165-1	Tape, 1in case	3	$12.50	$37.50
5	984567-12	Staples, wire, box	10	$1.00	$10.00
6	091344-5	Pens, box red felt tip	5	$4.50	$22.50
7	21457-3	Mousepad, blue	10	$0.50	$5.00
		Tax			$47.95
		Total Due			$527.45

View Ken Holman's work at: http://www.CraneSoftwrights.com/o/

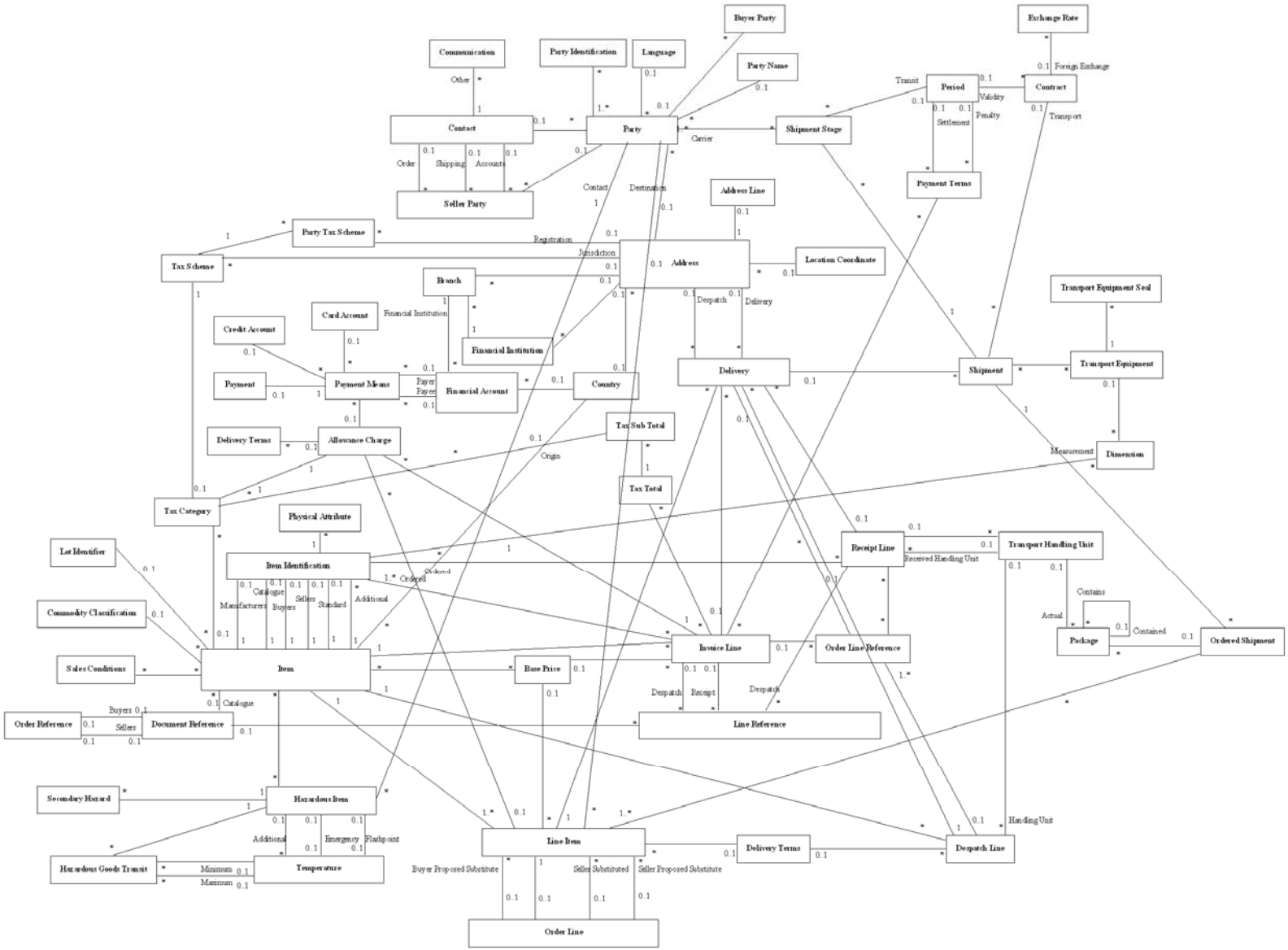
Order-to-Invoice Activity



©Mark Crawford, LMI Government Consulting, April 2005

UBL Document Component Model

LMI

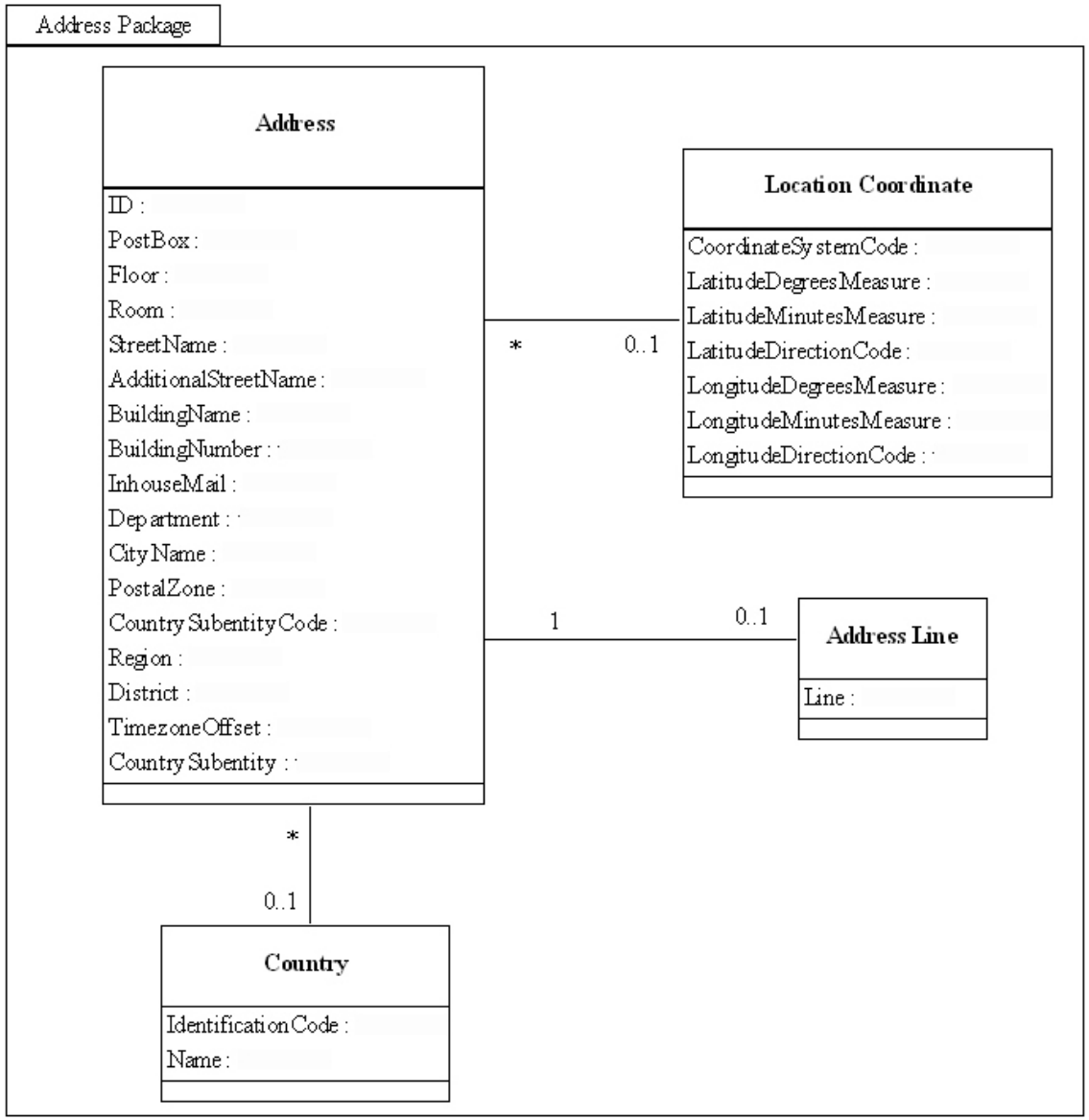


©Mark Craw

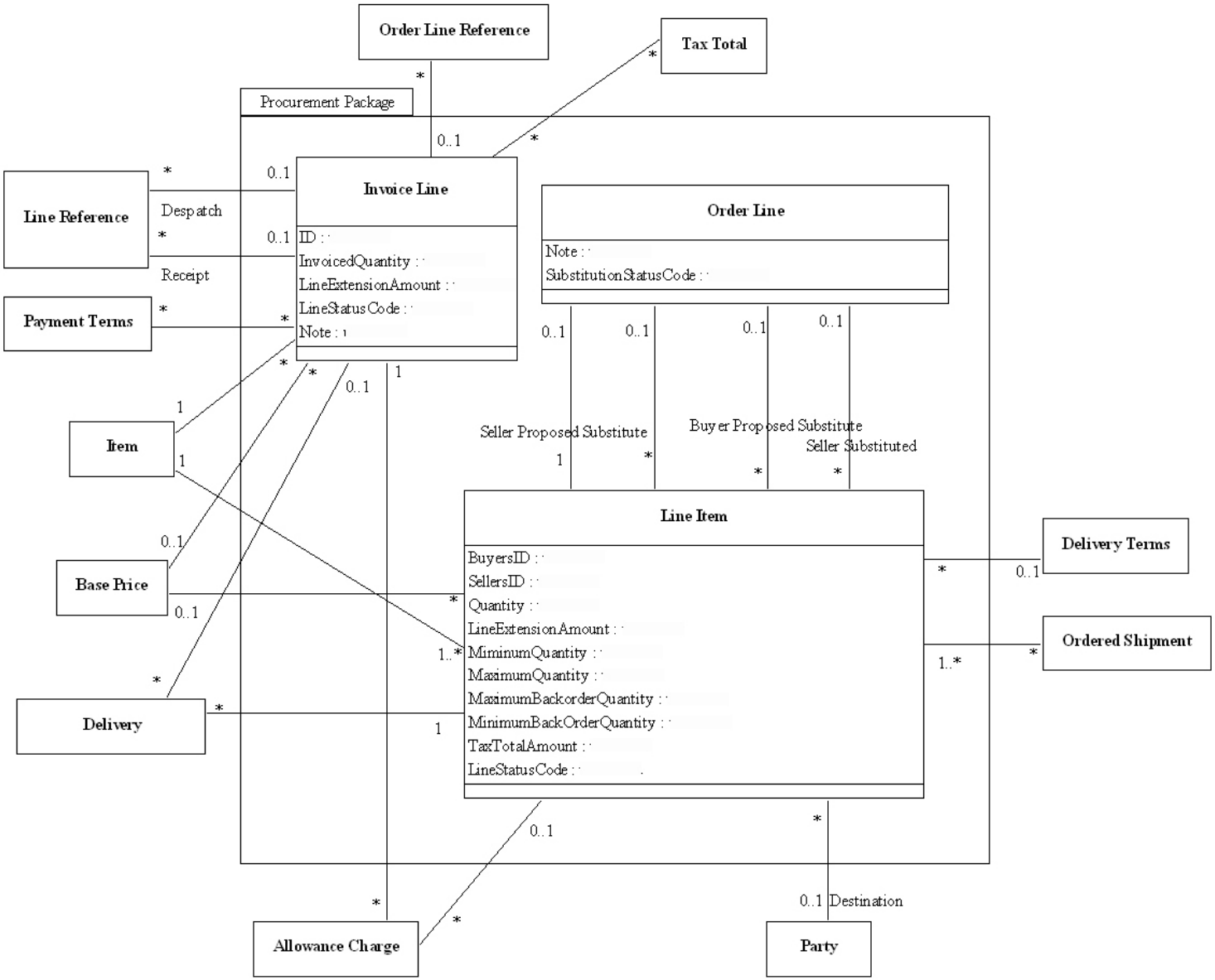
UBL Component Packages

- Address Contract
- Delivery
- Document Reference
- Hazardous Item
- Item
- Party
- Payment
- Procurement
- Tax

Address Package

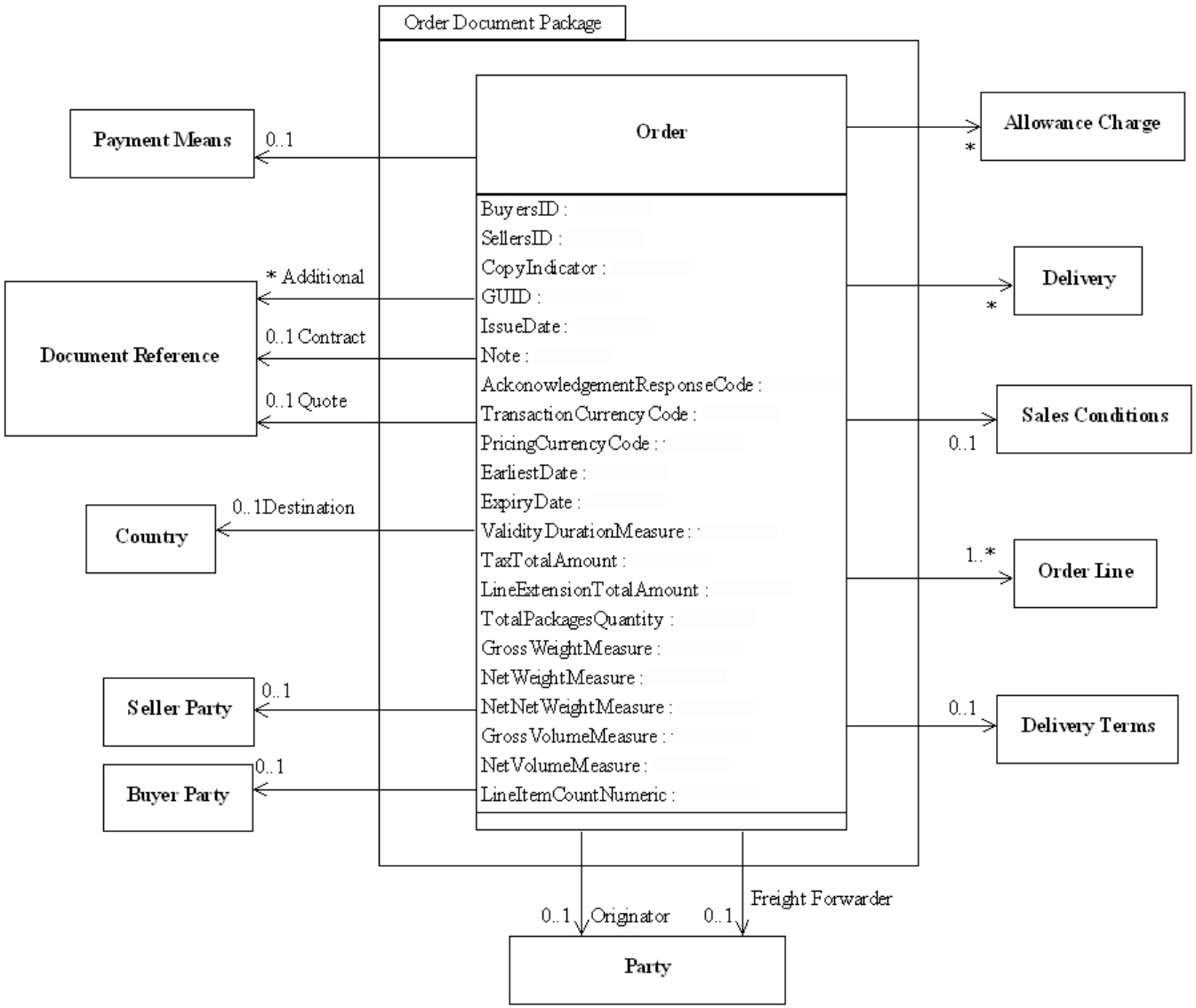


Procurement Package



©Mark Crawford, L

Order Document Package



©Mark Crawford, LMI Government Consulting, April 2005

Order Document Spreadsheet Snippet

UBL Name	Dictionary Entry Name
Order	Order. Details
BuyersID	Order. Buyers_ Identifier. Identifier
SellersID	Order. Sellers_ Identifier. Identifier
CopyIndicator	Order. Copy. Indicator
GUID	Order. Globally Unique_ Identifier. Identifier
IssueDate	Order. Issue Date. Date
Note	Order. Note. Text
AcknowledgementResponseCode	Order. Acknowledgement Response. Code
TransactionCurrencyCode	Order. Transaction Currency. Code
PricingCurrencyCode	Order. Pricing Currency. Code
EarliestDate	Order. Earliest Date. Date
ExpiryDate	Order. Expiry Date. Date
ValidityDurationMeasure	Order. Validity Duration. Measure
TaxTotalAmount	Order. Tax Total. Amount
LineExtensionTotalAmount	Order. Line_ Extension Total. Amount
TotalPackagesQuantity	Order. Total_ Packages Quantity. Quantity
GrossWeightMeasure	Order. Gross_ Weight. Measure
NetWeightMeasure	Order. Net_ Weight. Measure
NetNetWeightMeasure	Order. Net Net_ Weight. Measure
GrossVolumeMeasure	Order. Gross_ Volume. Measure

©Mark Crawford, LMI Government Consulting, April 2005

Implementation Model

- XSD Schema
  - **Order**
  - **Order Response**
  - **Order Response Simple**
  - **Order Change**
  - **Order Cancellation**
  - **Despatch Advice**
  - **Receipt Advice**
  - **Invoice**

- Code List Schema
  - **Acknowledgement Response Code**
  - **Allowance Charge Reason Code**
  - **Channel Code**
  - **Chip Code**
  - **Country Identification Code**
  - **Currency Code**
  - **Document Status Code**
  - **Latitude Direction Code**
  - **Line Status Code**
  - **Longitude Direction Code**
  - **Operator Code**
  - **Payment Means Code**
  - **Substitution Status Code**

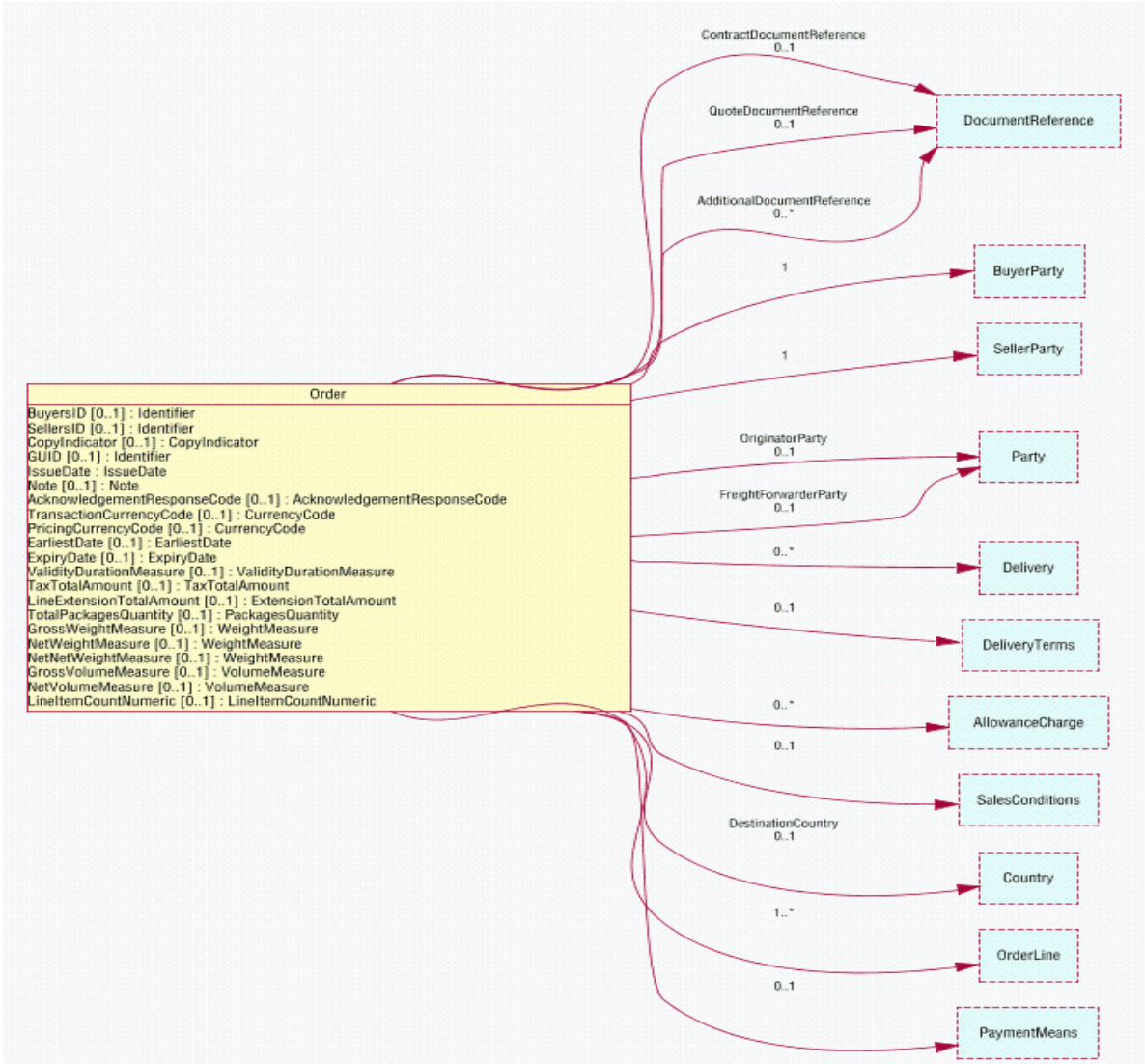
©Mark Crawford, LMI Government Consulting, April 2005

Implementation Model

- Reusable Component Class Diagrams
  - Autogenerated from XSD Schema using Ontogenics hyperModel*
    - Address
    - Contract
    - Despatch Line
    - Document Reference
    - ‘Hazardous Item
    - Item
    - Party
    - Payment
    - Procurement
    - Shipment
    - Tax

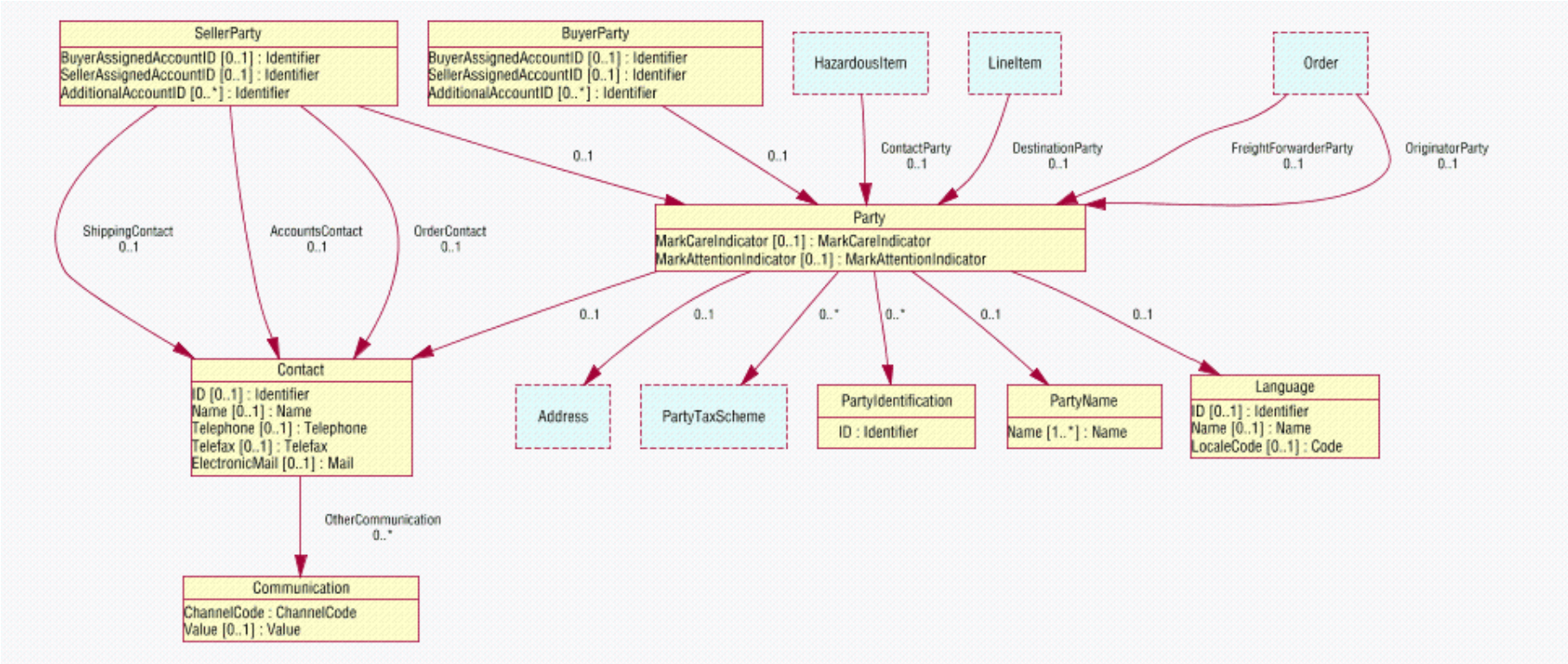
*View David Carlson’s work at: http://www.xmlmodeling.com/

Order Document Implementation



©Mark Crawford, LMI Government Consulting, April 2005

Party Relationship Implementation Diagram

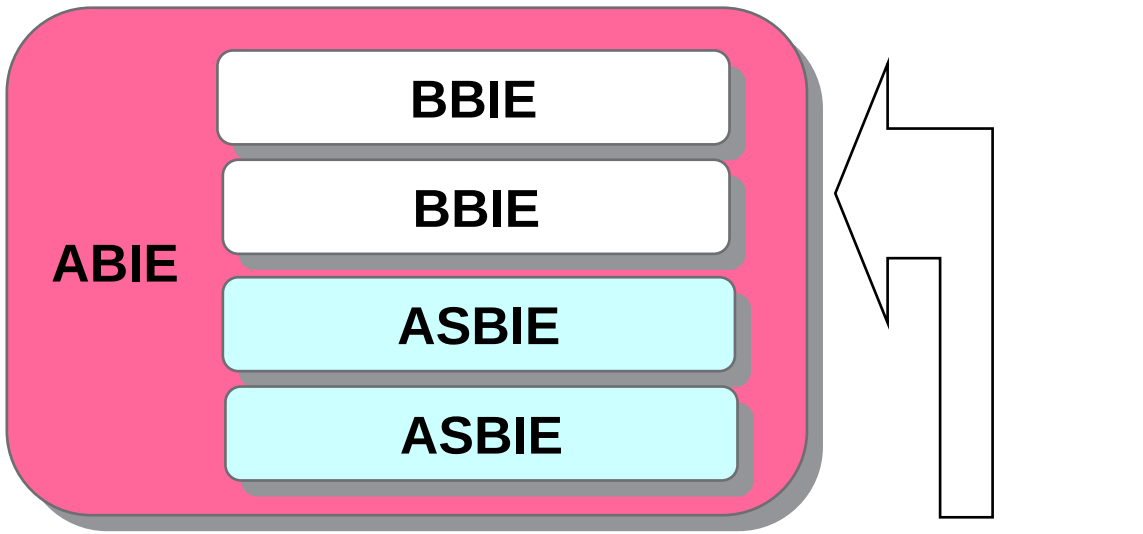


©Mark Crawford, LMI Government Consulting, April 2005

Creating UBL Schema

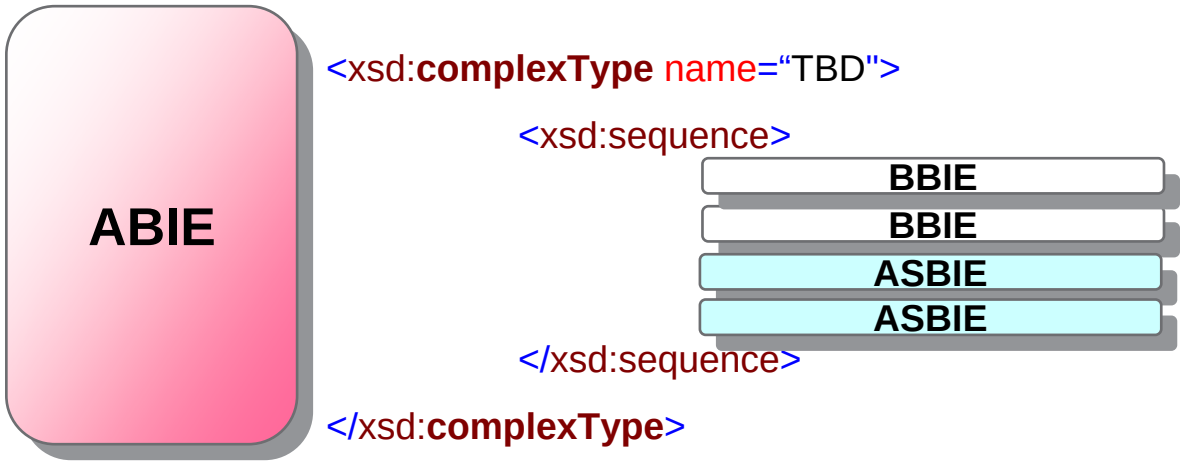
BIEs as Schema Content

- ABIE contains/aggregates ASBIEs and/or BBIEs



ABIEs as XML Schema

- ABIE → complexType
- ABIE aggregates elements (ASBIEs and BBIEs)



- BBIEs are **IN**trinsic to ABIE
- ASBIEs are **IN**trinsic to ABIE

See UBL Rule(s):
CTN1
ELN1

©Mark Crawford, LMI Government Consulting, April 2005

ABIEs as XML Schema: the UBL Rule(s)

LMI

[CTN1] A UBL xsd:complexType name based on an `ccts:AggregateBusinessInformationEntity` MUST be the `ccts:DictionaryEntryName` with the separators removed and with the "Details" suffix replaced with "Type".

pg. 48

[ELN1] A UBL global element name based on a `ccts:ABIE` MUST be the same as the name of the corresponding `xsd:complexType` to which it is bound, with the word "Type" removed.

pg. 50

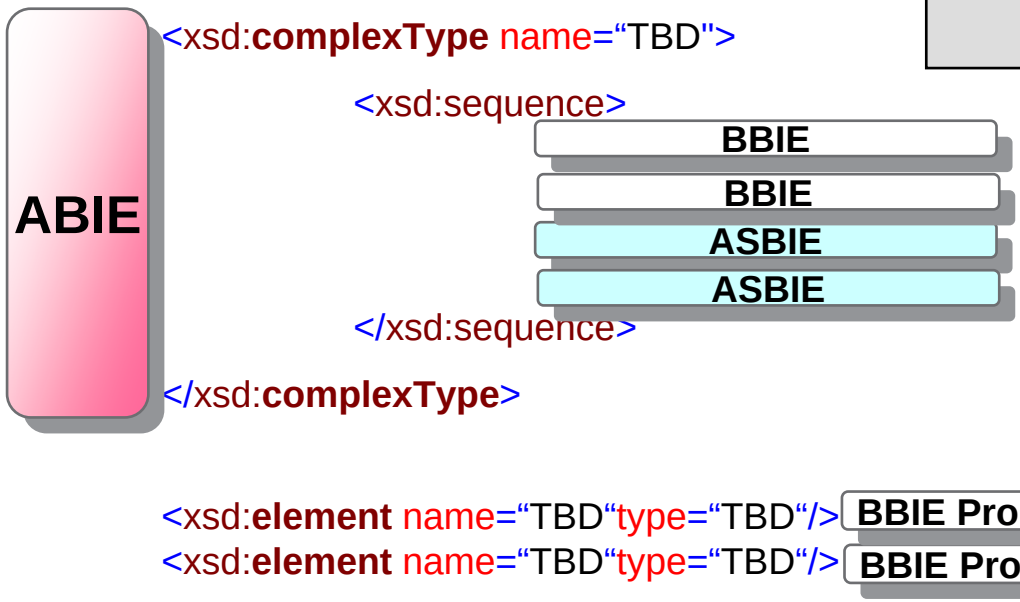
- 2 Schema representations for an ABIE
  - One is a complexType
  - The other is an element

©Mark Crawford, LMI Government Consulting, April 2005

BBIEs as XML Schema

- BBIE → ref within an ABIE
- BBIE Property → element declaration

See UBL Rule(s):
CTN2
ELN2



This distinction is important !

- BBIEs are **IN**trinsic to ABIE
- BBIE Properties are **EX**trinsic to ABIE
- BBIE Properties are linked to either UDT or QDT
- So, is a BBIE and a BBIE Property the same thing ? No.

©Mark Crawford, LMI Government Consulting, April 2005

BBIEs as XML Schema: the UBL Rule(s)

[CTN2]	A UBL xsd:complexType name based on a `ccts:BasicBusinessInformationEntityProperty` MUST be the `ccts:DictionaryEntryName` shared property term and its qualifiers and representation term of the shared `ccts:BasicBusinessInformation-Entity`, with the separators removed and with the "Type" suffix appended after the representation term.

pg. 48

[ELN2]	A UBL global element name based on an unqualified `ccts:BBIEProperty` MUST be the same as the name of the corresponding `xsd:complexType` to which it is bound, with the word "Type" removed.

pg. 51

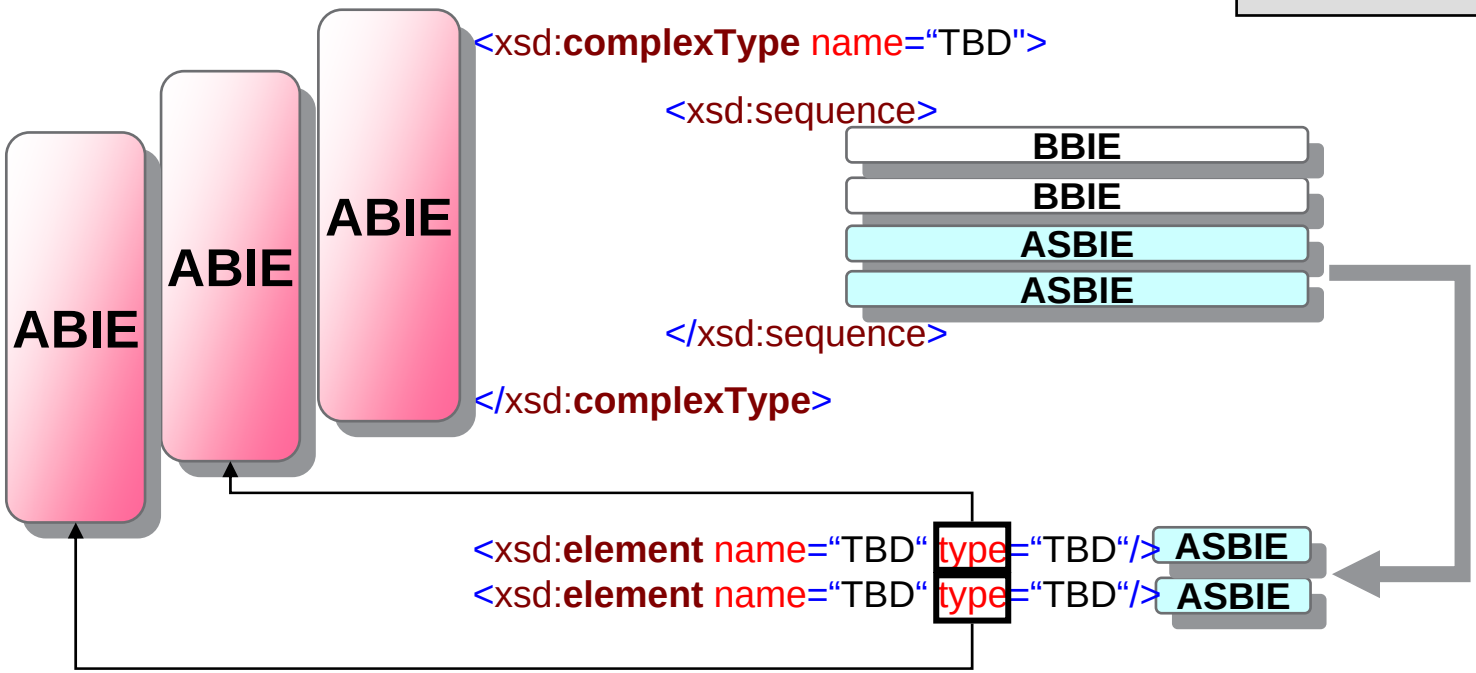
- 2 Schema representations for a BBIE
  - One is a complexType
  - The other is an element

©Mark Crawford, LMI Government Consulting, April 2005

ASBIEs as XML Schema

- Similar in structure to BBIEs
- ASBIE's → element refs

See UBL Rule(s):
ELN3
ELN4



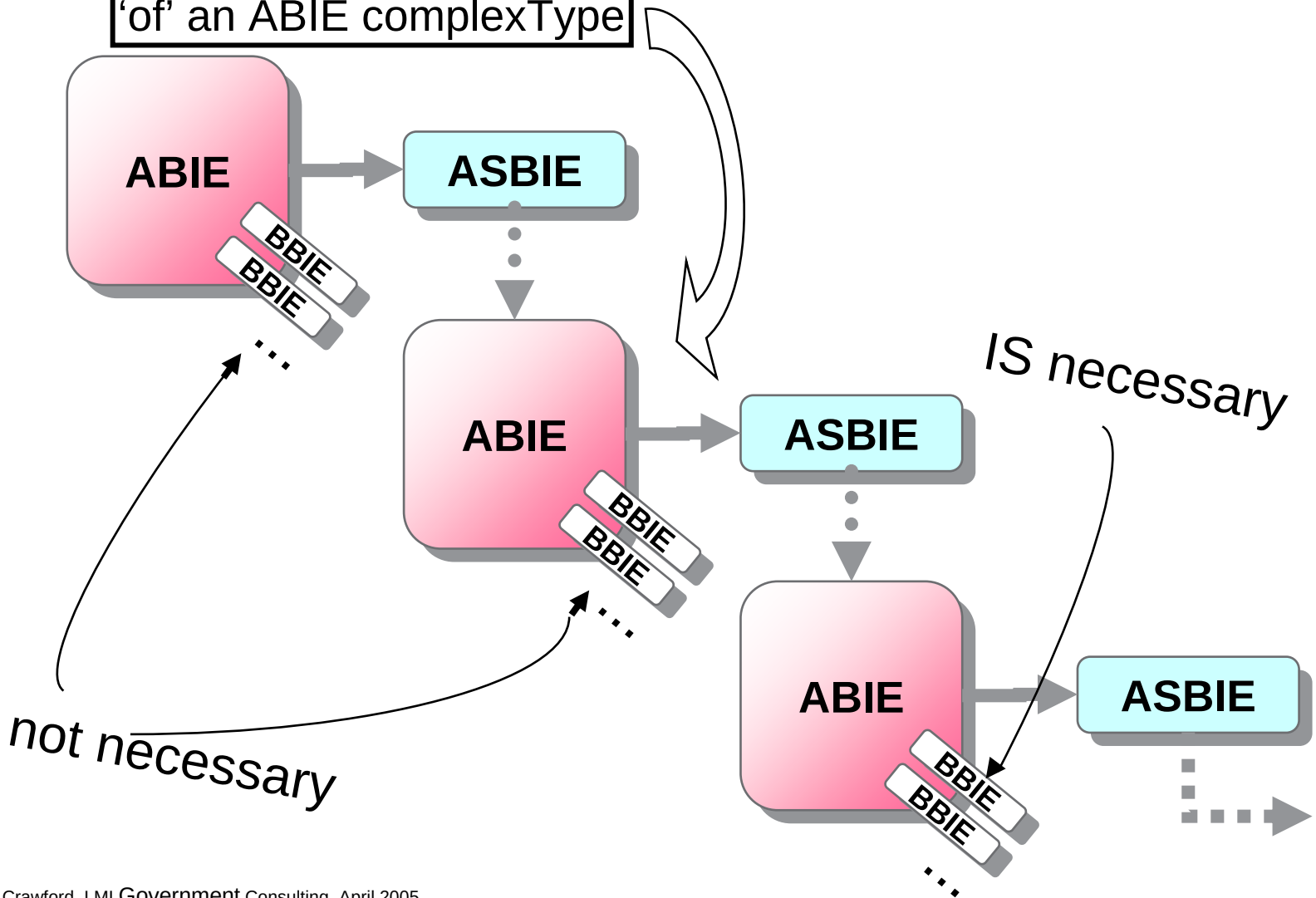
- Each ASBIE/element **ref**(erence)**s** an **EX**trinsic element that is of an ABIE complexType.

Critical !

©Mark Crawford, LMI Government Consulting, April 2005

ABIEs and ASBIEs

- Each ASBIE/element **ref**(erence)**s** an **EX**trinsic element that is 'of' an ABIE complexType



ASBIEs as XML Schema: the UBL Rule(s)

. . . is an association between two classes. As such, an element representing the `ccts:AssociationBusinessInformation Entity` does not have its own unique `xsd:ComplexType`. Instead, when an element representing a `ccts:AssociationBusinessInformationEntity` is declared, the element is bound to the `xsd:complexType` of its associated `ccts:Aggregate BusinessInformationEntity`.

ASBIEs as XML Schema: the UBL Rule(s)

[ELN3]A UBL global element name based on a qualified `ccts:ASBIE` MUST be the `ccts:ASBIE` dictionary entry name property term and its qualifiers; and the object class term and qualifiers of its associated `ccts:ABIE`. All `ccts:DictionaryEntryName` separators MUST be removed. Redundant words in the `ccts:ASBIE` property term or its qualifiers and the associated `ccts:ABIE` object class term or its qualifiers MUST be dropped.

pg. 52

- 1 Schema representation for an ASBIE

©Mark Crawford, LMI Government Consulting, April 2005

Supporting Data Types

LMI

- Qualified Data Types (QDTs)
  - Derived from UDTs
  - With restrictions (on the Content Component or Supplementary Component)
- Unqualified Data Types (UDTs)
  - Derived from Core Component Types
  - With NO restrictions (on the Content Component or Supplementary Component)
- Let’s look at some specific examples...

QDT – Review of Syntax Neutral

- Based on Unqualified Data Types (UDT)
- Specialization of UDT

CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Qualifier	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type
Mailing_ Address. Details	ABIE	Mailing	Address		Details					
Mailing_ Address. Line One. Text	BBIE	Mailing	Address		LineOne	Text		TextType	Text. Type	Text
Mailing_ Address. Line Two. Text	BBIE	Mailing	Address		LineTwo	Text		TextType	Text. Type	Text
Mailing_ Address. City Name. Text	BBIE	Mailing	Address		CityName	Text		TextType	Text. Type	Text
Mailing_ Address. Postcode. Code	BBIE	Mailing	Address		Postcode	Code	Postal_	CodeType	Postcode_ Code.	Code

UDT Code Type

is derived from

QDT

Code. Type

Postal_ Code. Type

©Mark Crawford, LMI Government Consulting, April 2005

QDT – Schema Syntax Specific

- BBIE Properties are linked to either UDT or QDT
- When linked to **QDT**... it's b/c of some restriction(s)

```
<xsd:complexType name="PostalCodeType">
        <xsd:annotation>
                <xsd:documentation>
                <ccts:ComponentType>TBD</ccts:ComponentType>
                <ccts:DictionaryEntryName>TBD</ccts:DictionaryEntryName>
                <ccts:Definition>TBD</ccts:Definition>
                <ccts:ObjectClass>TBD</ccts:ObjectClass>
                <ccts:RepresentationTerm>TBD</ccts:RepresentationTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
                <xsd:restriction base="udt:CodeType">
                        <xsd:minLength value="1"/>
                        <xsd:maxLength value="5"/>
                </xsd:restriction>
        </xsd:simpleContent>
</xsd:complexType>
```

©Mark Crawford, LMI Government Consulting, April 2005

UDT – Review of Syntax Neutral

- ‘Predefined’ Schema Module
- Based Directly on Core Component Types (CCTs)
- Used as the basis for Qualified Data Types

CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Qualifier	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type
Mailing_ Address. Details	ABIE	Mailing	Address		Details					
Mailing_ Address. Line One. Text	BBIE	Mailing	Address		LineOne	Text		TextType	Text. Type	Text
Mailing_ Address. Line Two. Text	BBIE	Mailing	Address		LineTwo	Text		TextType	Text. Type	Text
Mailing_ Address. City Name. Text	BBIE	Mailing	Address		CityName	Text		TextType	Text. Type	Text
Mailing_ Address. Postcode. Code	BBIE	Mailing	Address		Postcode	Code	Postalcode	CodeType	Postcode_ Code.	Code

Rep Term & Normalized CCT

is derived from

UDT

Text

Text. Type

UDT – Schema Syntax Specific

- BBIE Properties are linked to either UDT or QDT
- When linked to **UDT**...
  - restriction(s) may NOT exist
  - extensions should be used

See UBL Rule(s):
CTD3
CTD4
CTD5

```
<xsd:complexType name="DepartmentNameType">
        <xsd:annotation>
                <xsd:documentation>
                <ccts:ComponentType></ccts:ComponentType>
                <ccts:DictionaryEntryName></ccts:DictionaryEntryName>
                <ccts:Definition></ccts:Definition>
                <ccts:ObjectClass></ccts:ObjectClass>
                <ccts:RepresentationTerm></ccts:RepresentationTerm>
                </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
                <xsd:extension base="udt:TextType"/>
        </xsd:simpleContent>
</xsd:complexType>
```

©Mark Crawford, LMI Government Consulting, April 2005

BBIE/UDT/QDT as XML Schema: the UBL Rule(s)

LMI

5.1.3.2 Basic Business Information Entities

Basic Business Information Entities (BBIEs), in accordance with the Core Components Technical Specification, always have a primary representation term, and may have secondary representation terms, which describes their structural representation. These representation terms are expressed in the UBL Model as Unspecialised Datatypes bound to a Core Component Type that describes their structure. In addition to the unspecialised Datatypes defined in CCTS, UBL has defined a set of specialised Datatypes that are derived from the CCTS unqualified Datatypes.There are a set of rules concerning the way these relationships are expressed in the UBL XML library. As discussed above, BBIE properties are represented with complex types. Within these are simpleContent elements that extend the Datatypes.

[CTD3]	Every `ccts:BBIEProperty xsd:complexType` definition content model MUST use the `xsd:simpleContent` element.

[CTD4]	Every `ccts:BBIEProperty ComplexType` content model `xsd:simpleContent` element MUST consist of an `xsd:extension` element.

[CTD5]	Every `ccts:BBIEProperty xsd:complexType` content model `xsd:base` attribute value MUST be the `ccts:CCT` of the unspecialised or specialised UBL Datatype as appropriate.

So, what does a UDT look like ?

```
<xsd:complexType name="TextType">
    <xsd:annotation> ...
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="languageID" type="clm5639:LanguageCodeContentType“ use="optional">
        <xsd:attribute name="languageLocaleID" type="xsd:normalizedString" use="optional">
      </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
```

- UDTs are based on Schema ‘built-in’ data types

See UBL Rule(s):
CTD7 - CTD12

UDT as XML Schema: the UBL Rule(s)

LMI

5.1.3.3.1 Unspecialised Datatypes

The `ccts:UnspecialisedDatatypes` reflect the instantiation of the `ccts:CoreComponentTypes`. Each `ccts:UnspecialisedDatatype` declaration is based on its corresponding qualified `ccts:CoreComponentType` and represents either a primary or secondary representation term.

[CTD7]	Every unspecialised Datatype must be based on a ccts:CCT represented in the CCT schema module, and must represent an approved primary or secondary representation term identified in the CCTS.
[CTD8]	Each unspecialised Datatype xsd:complexType must be based on its corresponding CCT xsd:complexType.
[CTD9]	Every unspecialised Datatype that represents a primary representation term whose corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.
[CTD10]	Every unspecialised Datatype that represents a secondary representation term whose corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.
[CTD11]	Each unspecialised Datatype xsd:complexType definition must contain one xsd:simpleContent element.
[CTD12]	The unspecialised Primary Representation Term Datatype xsd:complexType definition xsd:simpleContent element must contain one xsd:restriction element with an xsd:base attribute whose value is equal to the corresponding cct:complexType

BIEs – A Specific Example

- Now.... let's look at some specific ABIE/BBIE/ASBIE examples !
- We'll be putting all of this to use in the 'Step By Step' session
- Between now and then we'll build on these ABIE/BBIE/ASBIE concepts by outlining their places in the Schema hierarchy
  - in the 'Schema Modularity' session

Example Source

- Locate the ‘Supplier Organization’ ABIE in your course spreadsheet
  - value for (column ‘H’) is: ‘Supplier_ Organization. Details’

XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Term	Property Term	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type (CCT)	Associated Object Class Qualifier	Associated Object Class Term
SupplierOrganizationType	**Supplier_ Organization. Details**	**ABIE**	**Organization**		**Details**							
NameType	Supplier_ Organization. Name. Text	**BBIE**	Organization		Name	Text		TextType	Text. Type	Text		
DepartmentNameType	Supplier_ Organization. Department_ Name. Text	**BBIE**	Org			Text		TextType	Text. Type	Text		
DepartmentIDType	Supplier_ Organization. Department_ Identification. Identifier	**BBIE**	Organization	Department	Identification	Identifier	Department	IdentifierType	Department_ Identifier. Type	Identifier		
	Supplier_ Organization. Official_ Contact. Mailing_ Address	**ASBIE**	Org								Mailing	Address
	Supplier_ Organization. Availability. Inventory_ Item	**ASBIE**	Organization		Availability						Inventory	Item

Complex Type

Element Ref (BBIE), Element (BBIE Prop)

Element Ref, Element

©Mark Crawford, LMI Government Consulting, April 2005

ABIE: Supplier_ Organization. Details

- ABIE → complexType

XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type (CCT)	Associated Object Class Qualifier	Associated Object Class Term
SupplierOrganizationType	Supplier_ Organization. Details	ABIE	Organization		Details							

ABIE

<xsd:**complexType** name="SupplierOrganizationType">

<xsd:annotation/> ... omitted for brevity

<xsd:sequence>

BBIE

BBIE

BBIE

ASBIE

ASBIE

</xsd:sequence>

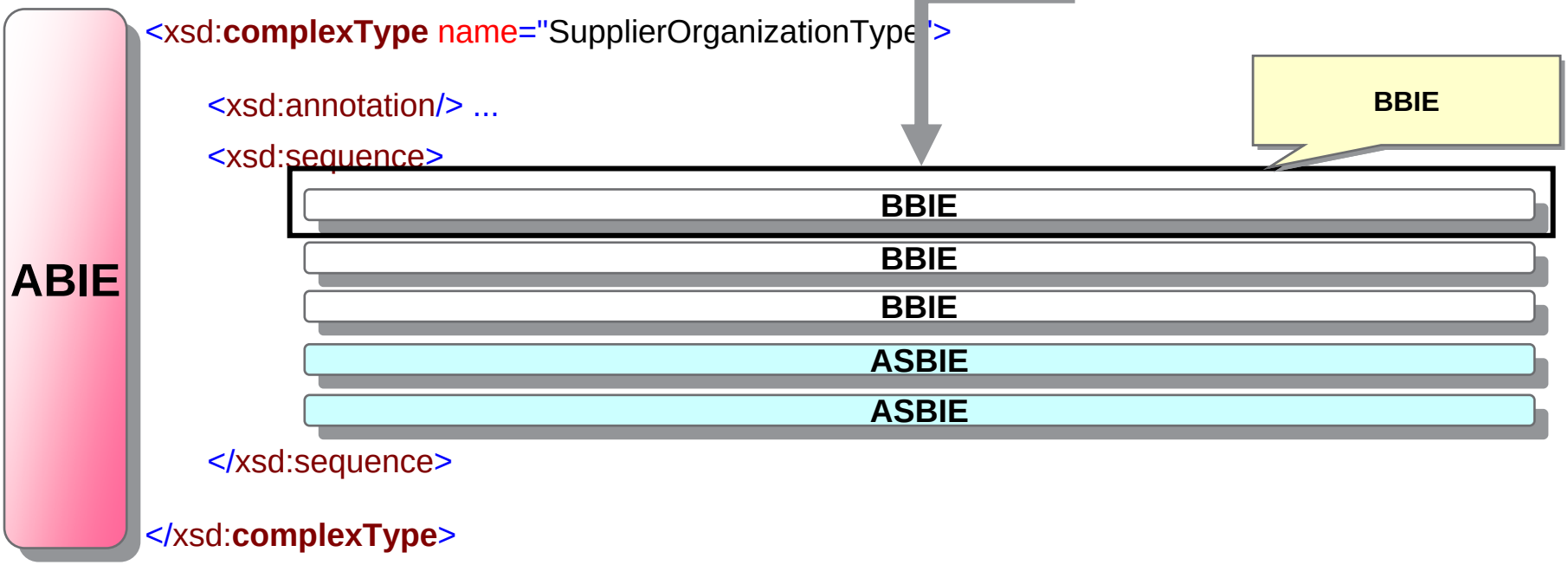
</xsd:**complexType**>

©Mark Crawford, LMI Government Consulting, April 2005

ABIE: Supplier_ Organization. Details

- Now, let's look at the 1st of these 3 BBIEs
- And its associated BBIE Property, of course

Supplier_ Organization. Name. Text



BBIE: Supplier_ Organization. Name. Text

XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type (CCT)	Associated Object Class Qualifier	Associated Object Class Term
NameType	Supplier_ Organization. Name. Text	**BBIE**	Organization		Name	Text		TextType	Text. Type	Text		

```
<xsd:element name="Name" type="cbc:NameType">

<xsd:complexType name="NameType">
        <xsd:simpleContent>
                <xsd:extension base="udt:TextType"/>
        </xsd:simpleContent>
</xsd:complexType>
```

BBIE Property

here's the UDT from earlier

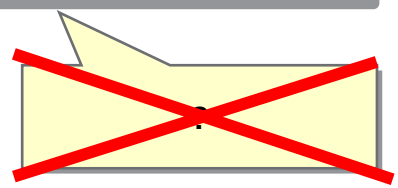
ABIE: Supplier_ Organization. Details

- Now let's back up to the ABIE again
- And look at the 1st of the 2 ASBIEs

Supplier_ Organization. Official_ Contact. Mailing_ Address

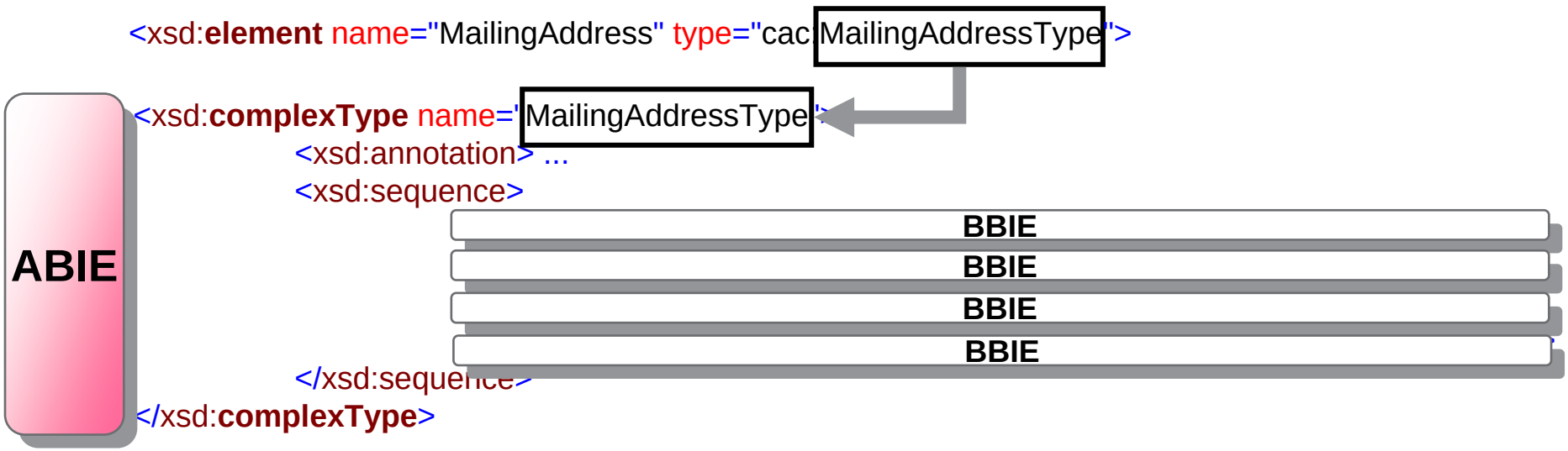
ABIE

```
<xsd:complexType name="SupplierOrganizationType">
    <xsd:annotation/> ...
    <xsd:sequence>
        BBIE
        BBIE
        BBIE
        ASBIE
        ASBIE
    </xsd:sequence>
</xsd:complexType>
```



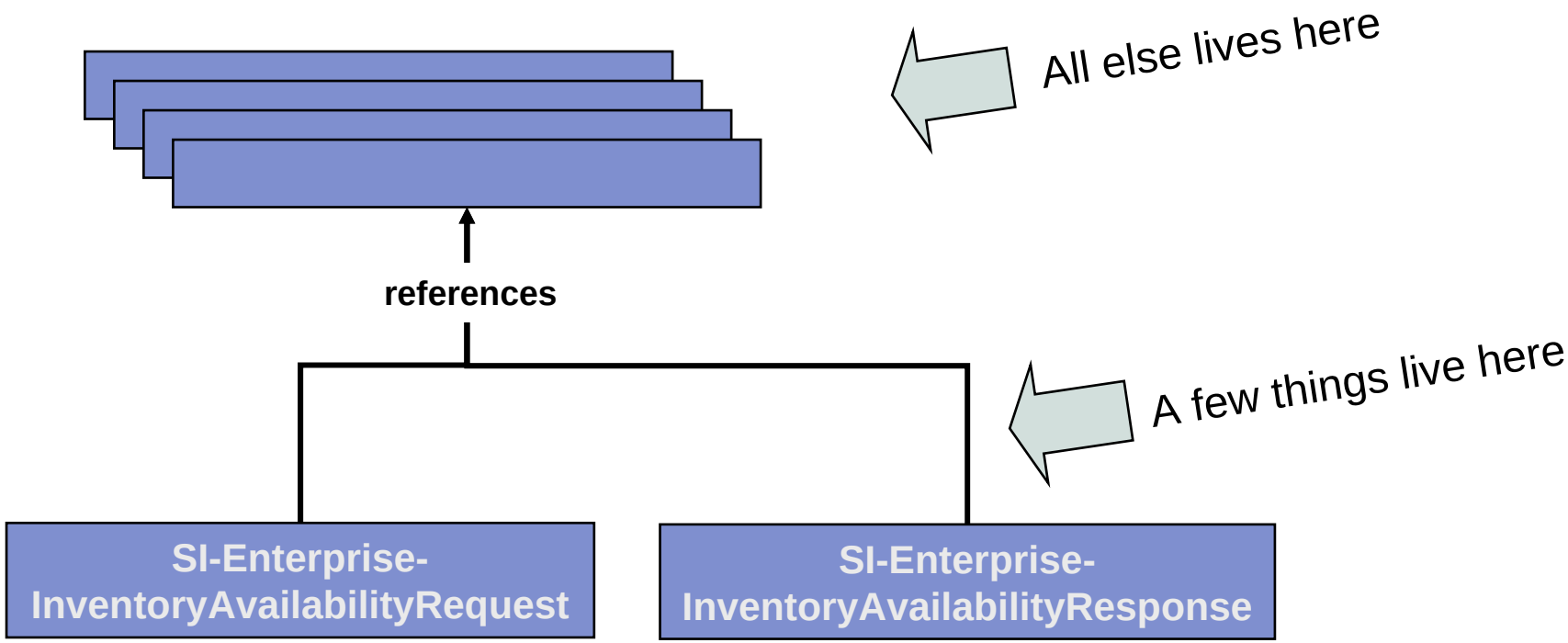
ASBIE: Supplier_ Organization. Official_ Contact. Mailing_ Address

XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type (CCT)	Associated Object Class Qualifier	Associated Object Class Term
	Supplier_ Organization. Official_ Contact. Mailing_ Address	**ASBIE**	Organization	Official	Contact						Mailing	Address



Message Assembly

- Message Assemblies represent the base entities (ie., root elements) that are passed in an XML message

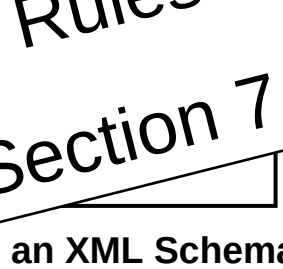


©Mark Crawford, LMI Government Consulting, April 2005

Schema Documentation – At a Glance

- Schema Documentation
  - Not to be confused with 'U
  - Then what is it?

The UBL NDR Provides a comprehensive set of Schema Documentation Rules that instantiate the storage rules from Section 7 of ISO 1500-5



an XML Schema

paper documentation

- Let's look at this in a Schema file ?...

Schema Documentation

- Much of the XSD content in these course slides has omitted annotation / documentation (for brevity)
- Are ‘annotations’ and ‘documentation’ the same thing ? No.

```
<xsd:complexType name="SupplierOrganizationType">
    <xsd:annotation>
      <xsd:documentation>
        <ccts:ComponentType>ABIE</ccts:ComponentType>
        <ccts:DictionaryEntryName>Supplier_ Organization. Details</ccts:DictionaryEntryName>
        <ccts:Definition>This holds all pertinent information relating to a supplier organization.</ccts:Definition>
        <ccts:ObjectClass>Organization</ccts:ObjectClass>
      </xsd:documentation>
    </xsd:annotation>
    . . .
</xsd:complexType>
```

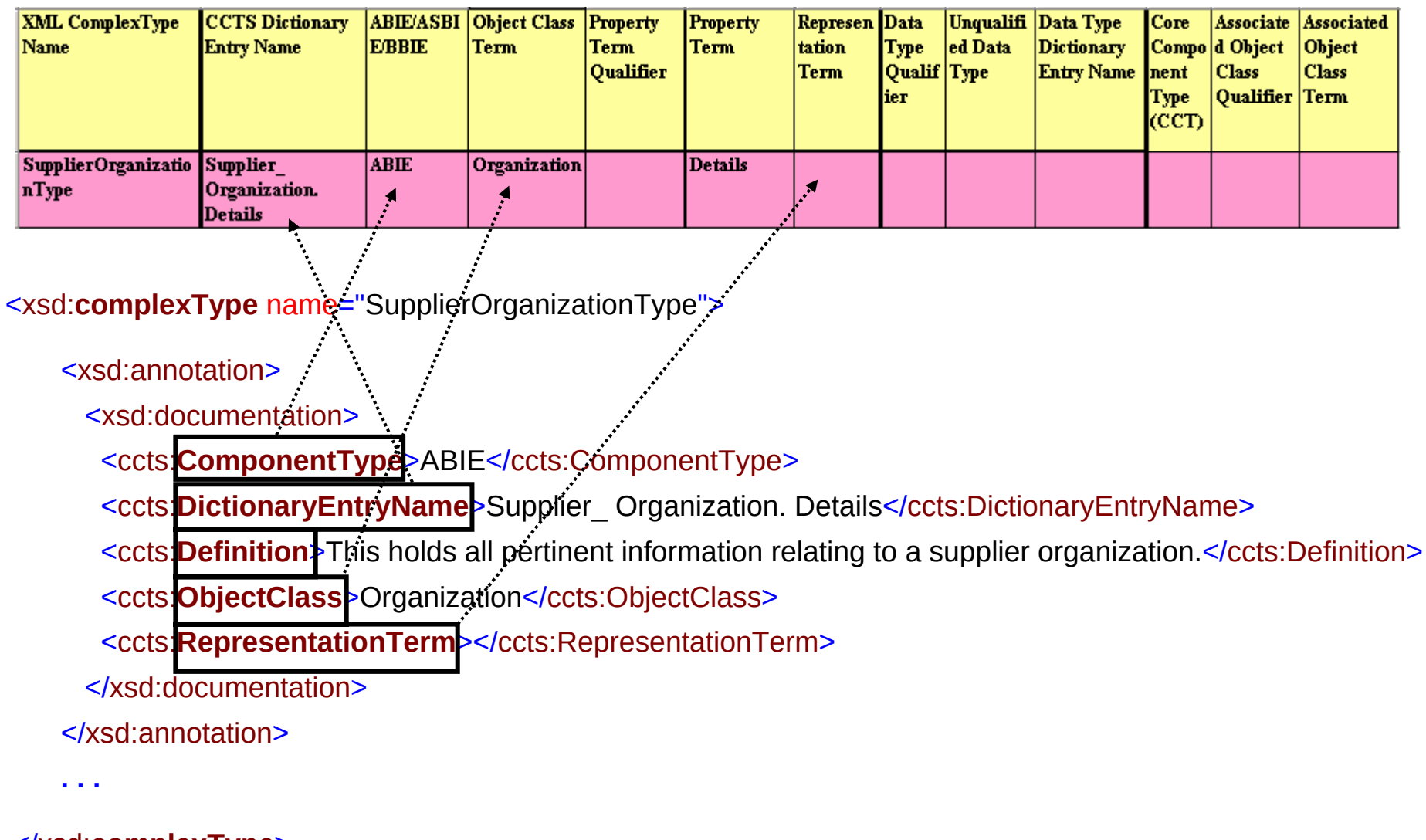
⇦ *here’s the ABIE from earlier*

- Let’s look at this mapped back to the spreadsheet...

©Mark Crawford, LMI Government Consulting, April 2005

XSD Documentation

XML ComplexType Name	CCTS Dictionary Entry Name	ABIE/ASBIE/BBIE	Object Class Term	Property Term Qualifier	Property Term	Representation Term	Data Type Qualifier	Unqualified Data Type	Data Type Dictionary Entry Name	Core Component Type (CCT)	Associated Object Class Qualifier	Associated Object Class Term
SupplierOrganizationType	Supplier_ Organization. Details	ABIE	Organization		Details							



©Mark Crawford, LMI Government Consulting, April 2005

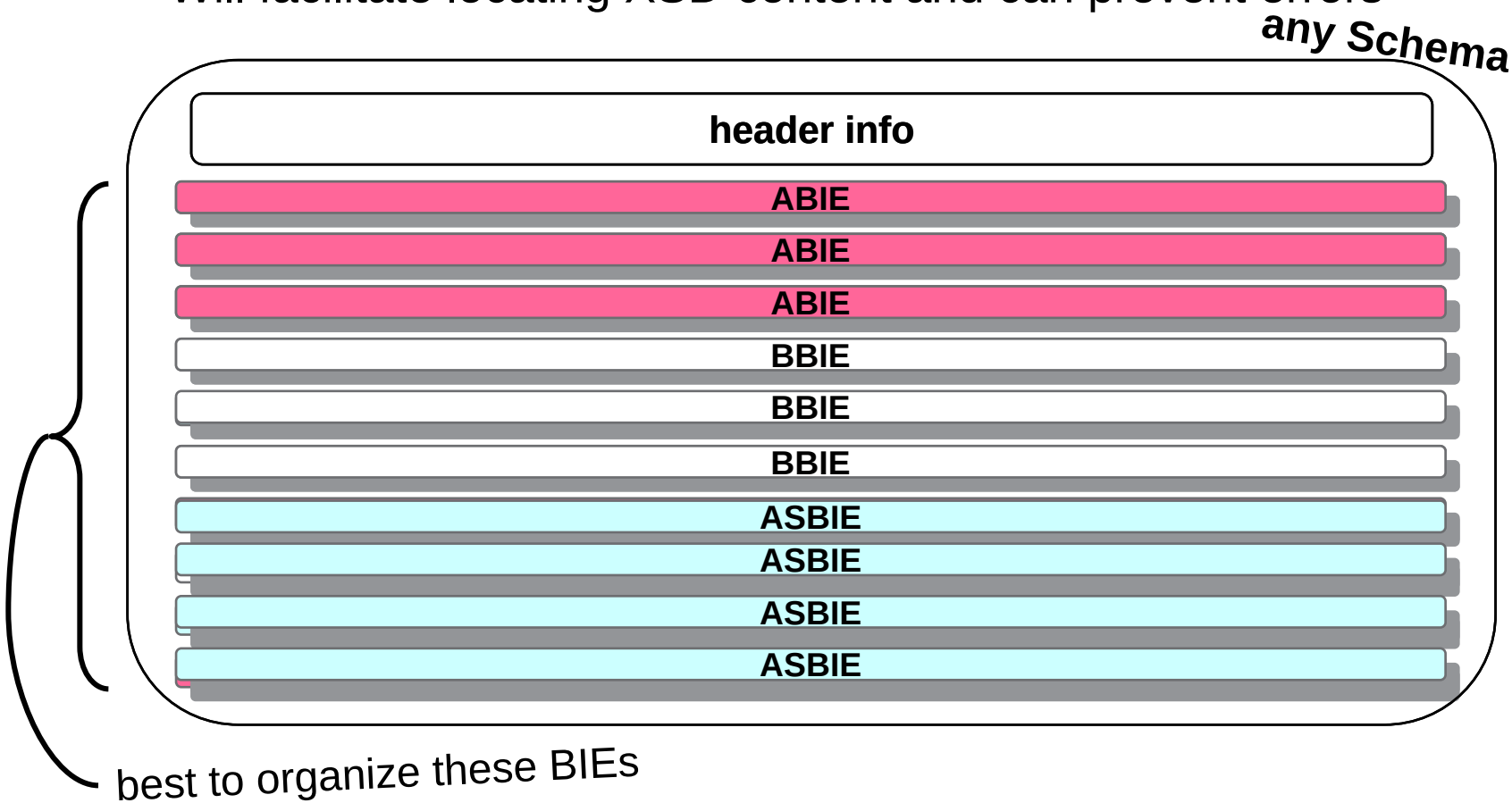
XSD Documentation: UBL NDR

The following rule describes the documentation requirements for each Aggregate Business Information Entity definition.

[DOC5]	The xsd:documentation element for every Aggregate Business Information Entity MUST contain a structured set of annotations in the following sequence and pattern: • ComponentType (mandatory): The type of component to which the object belongs. For Aggregate Business Information Entities this must be "ABIE". • DictionaryEntryName (mandatory): The official name of the Aggregate Business Information Entity . • Version (optional): An indication of the evolution over time of the Aggregate Business Information Entity. • Definition(mandatory): The semantic meaning of the Aggregate Business Information Entity. • ObjectClassQualifier (optional): The qualifier for the object class.
	• ObjectClass(mandatory): The Object Class represented by the Aggregate Business Information Entity. • AlternativeBusinessTerms (optional): Any synonym terms under which the Aggregate Business Information Entity is commonly known and used in the business.

Arranging the content

- Establish a consistent approach to arranging the content within ALL of your Schemas
- Will facilitate locating XSD content and can prevent errors



©Mark Crawford, LMI Government Consulting, April 2005

Referencing External Content

- By using the **xsd:include** and **xsd:import** directives
- Use these when the Schema content needed (ref'd) is located in another file / namespace

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema

targetNamespace="urn:us:com:supplyinventory:inventorydepartment:1.0“
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:us:com:supplyinventory:inventorydepartment:1.0"
xmlns:cac="urn:us:com:supplyinventory:enterprise:commonaggregatecomponents"
xmlns:ccts="urn:un:unece:uncefact:documentation:corecomponenttechnicalspecificati
on:2.0“

<xsd:import
namespace="urn:us:com:supplyinventory:enterprise:commonaggregatecomponents"
schemaLocation="SI-Enterprise-CommonAggregateComponents.xsd"/>

...
```

Schema root

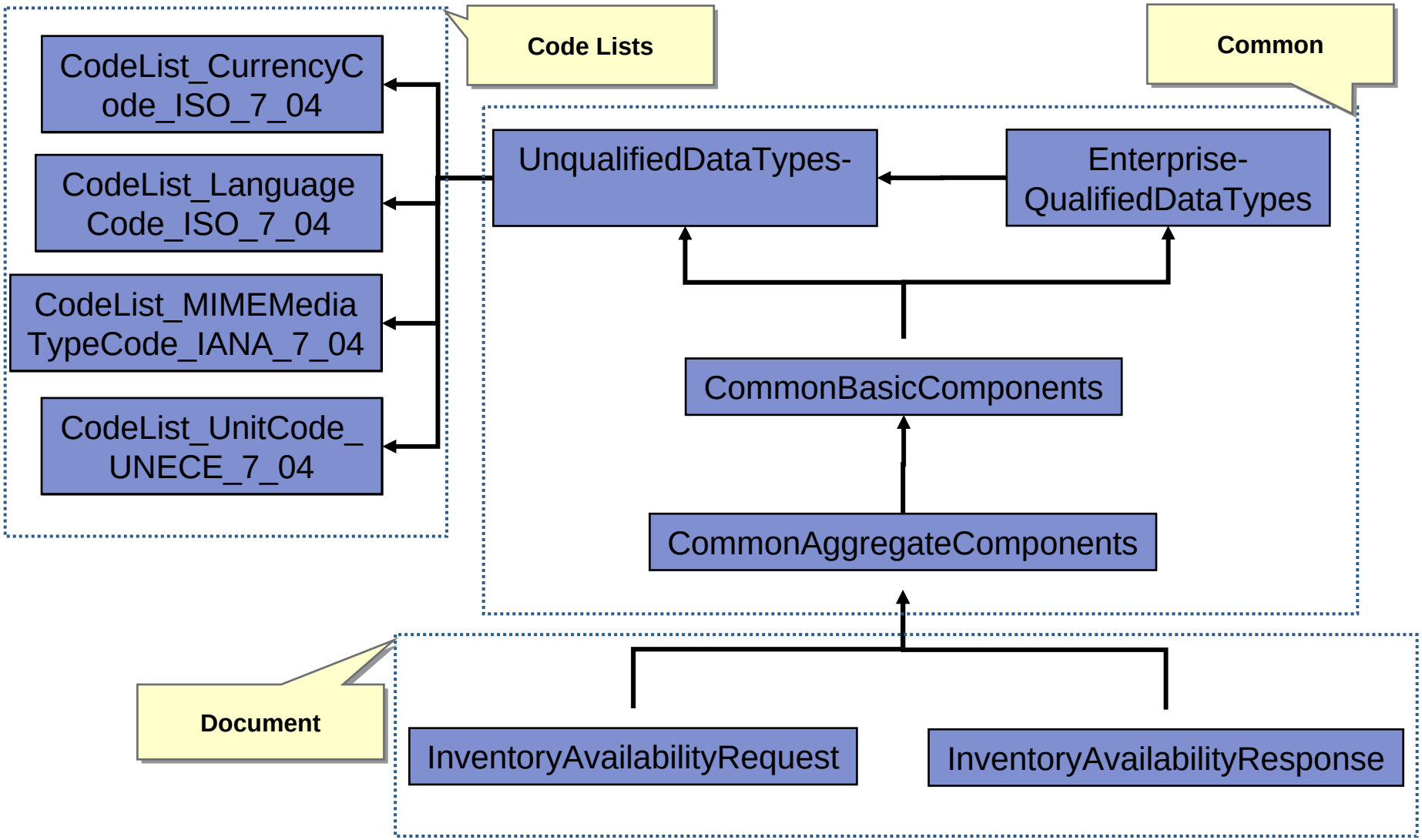
Namespaces

Import(s) / Include(s)

- This leads us into the next topic...

©Mark Crawford, LMI Government Consulting, April 2005

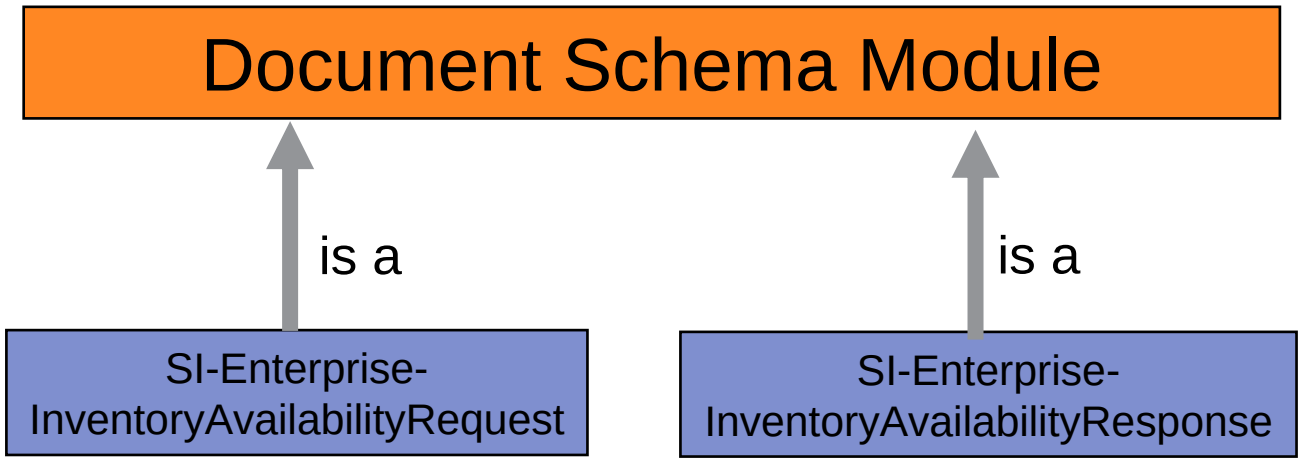
Schema Modularity



©Mark Crawford, LMI Government Consulting, April 2005

A word about the word ‘Document’

- A Document Schema represents the root level content definition
  - lowest or highest... depending on your proclivity



- Document does not denote / connote a ‘narrative’ document
- These XML ‘document’ Schemas define XML transactions for exchange between app servers

important terminology

©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header

Message Assembly

Document Schema Module

<xsd:schema targetNamespace="...>

ABIE placeholder

...

BBIE placeholder

...

ASBIE placeholder

...

...

</xsd:schema>

©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header

- Create the Schema root

- Define the targetNamespace
- Define the W3C XML Schema namespace
- Define all other namespace(s) *
- Define the default namespace

- Set the version

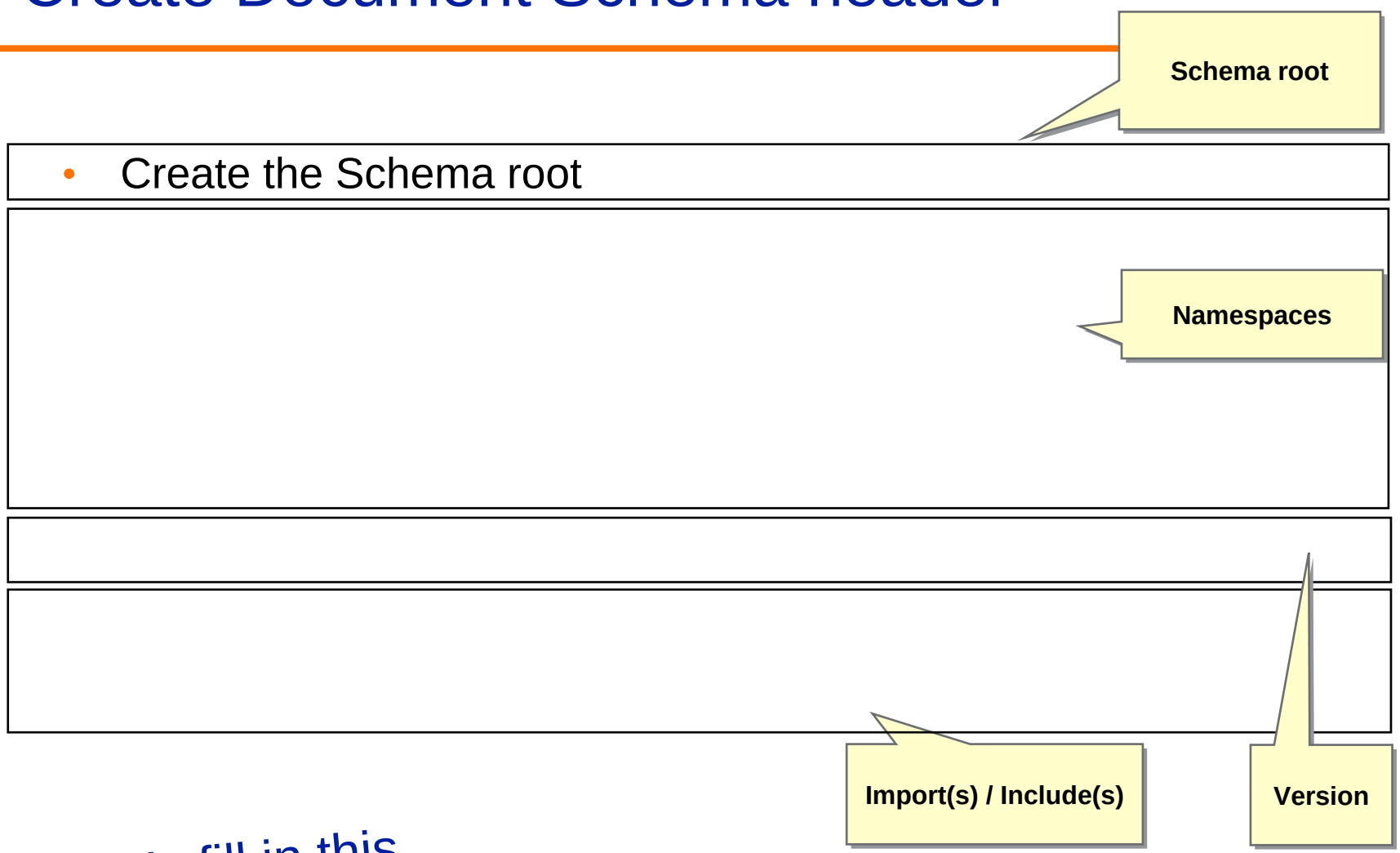
- Create imports *
- Create includes *

* when applicable

©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header

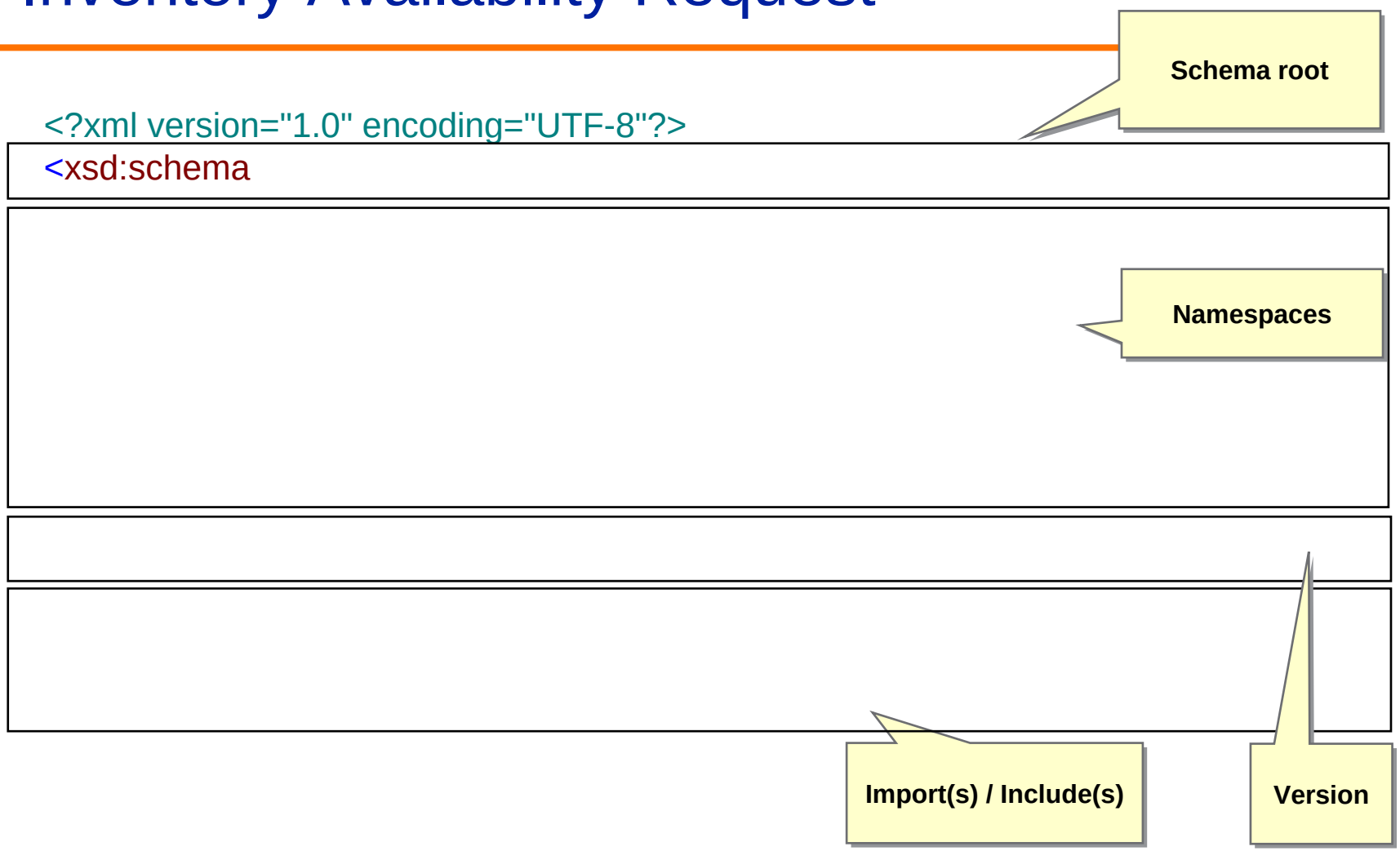
- Create the Schema root



Let's fill in this Schema header

©Mark Crawford, LMI Government Consulting, April 2005

Document Header for: Inventory Availability Request



©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header

LMI

- Create the Schema root

Schema root

- Define the targetNamespace
- Define the W3C XML Schema namespace
- Define all other namespace(s) *
- Define the default namespace

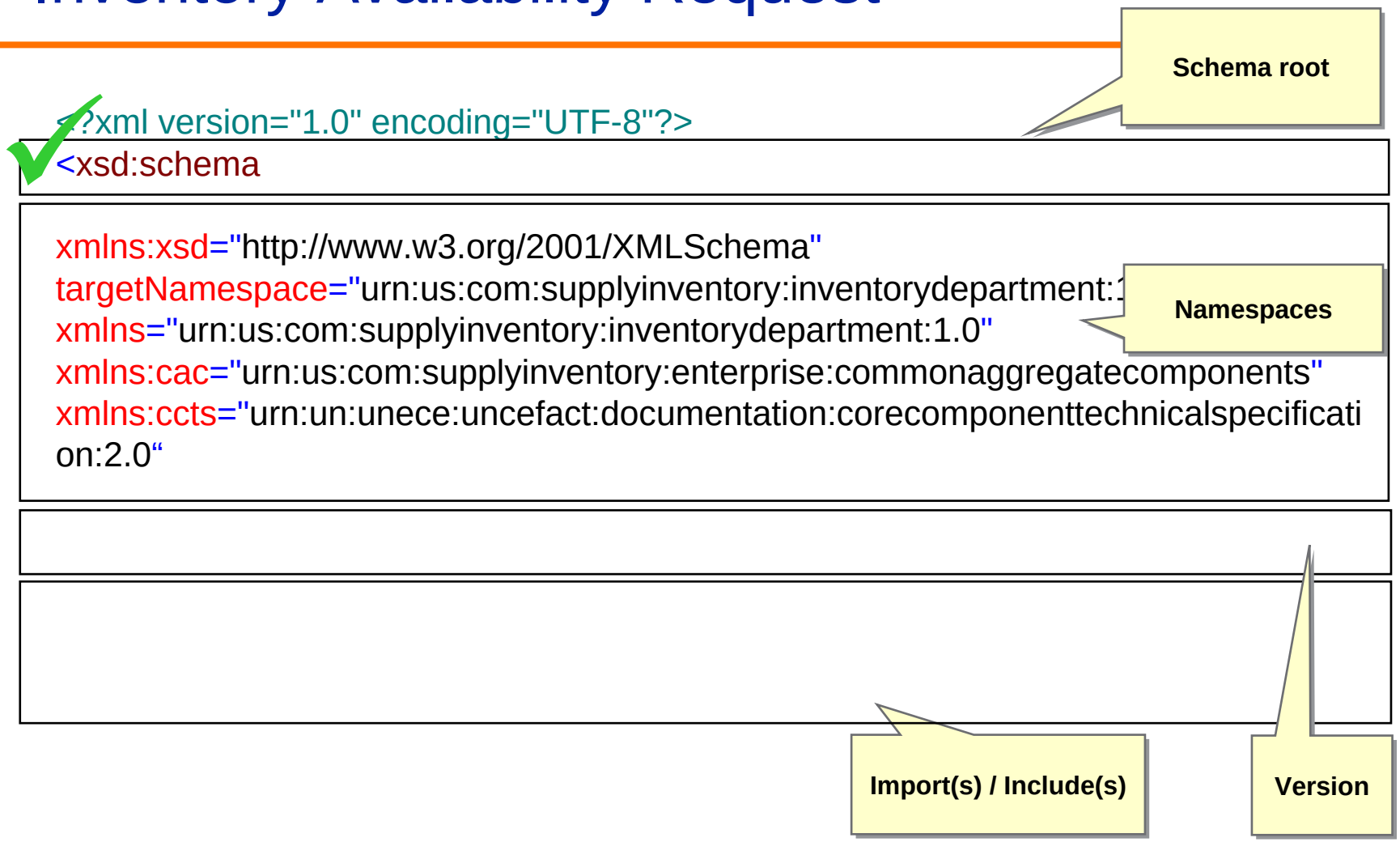
Namespaces

Import(s) / Include(s)

Version

Document Header for: Inventory Availability Request

LMI



©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header

LMI

- Create the Schema root

Schema root

- Define the targetNamespace

- Define the W3C XML Schema namespace
- Define all other namespace(s) *
- Define the default namespace

Namespaces

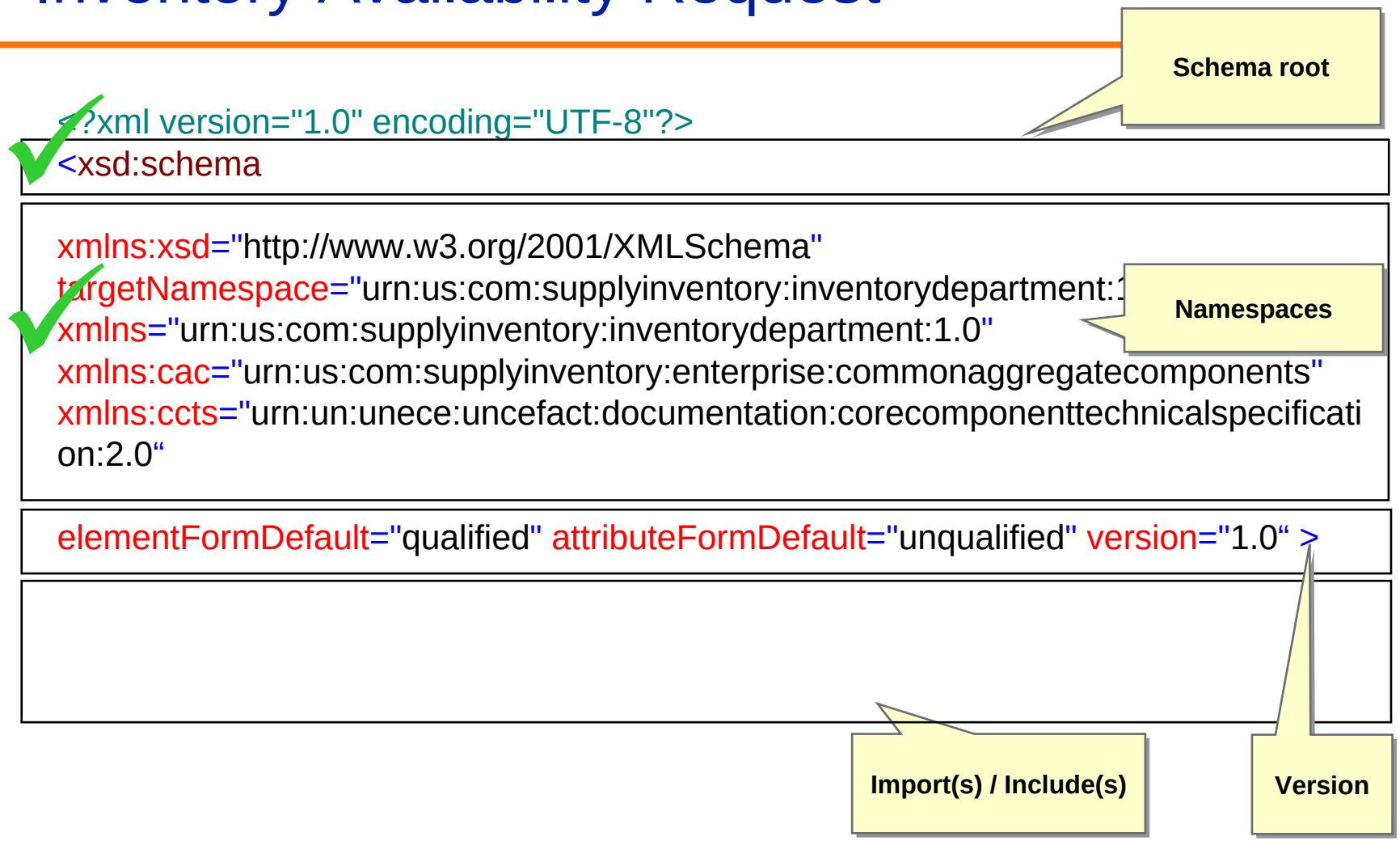
- Set the version

Version

Import(s) / Include(s)

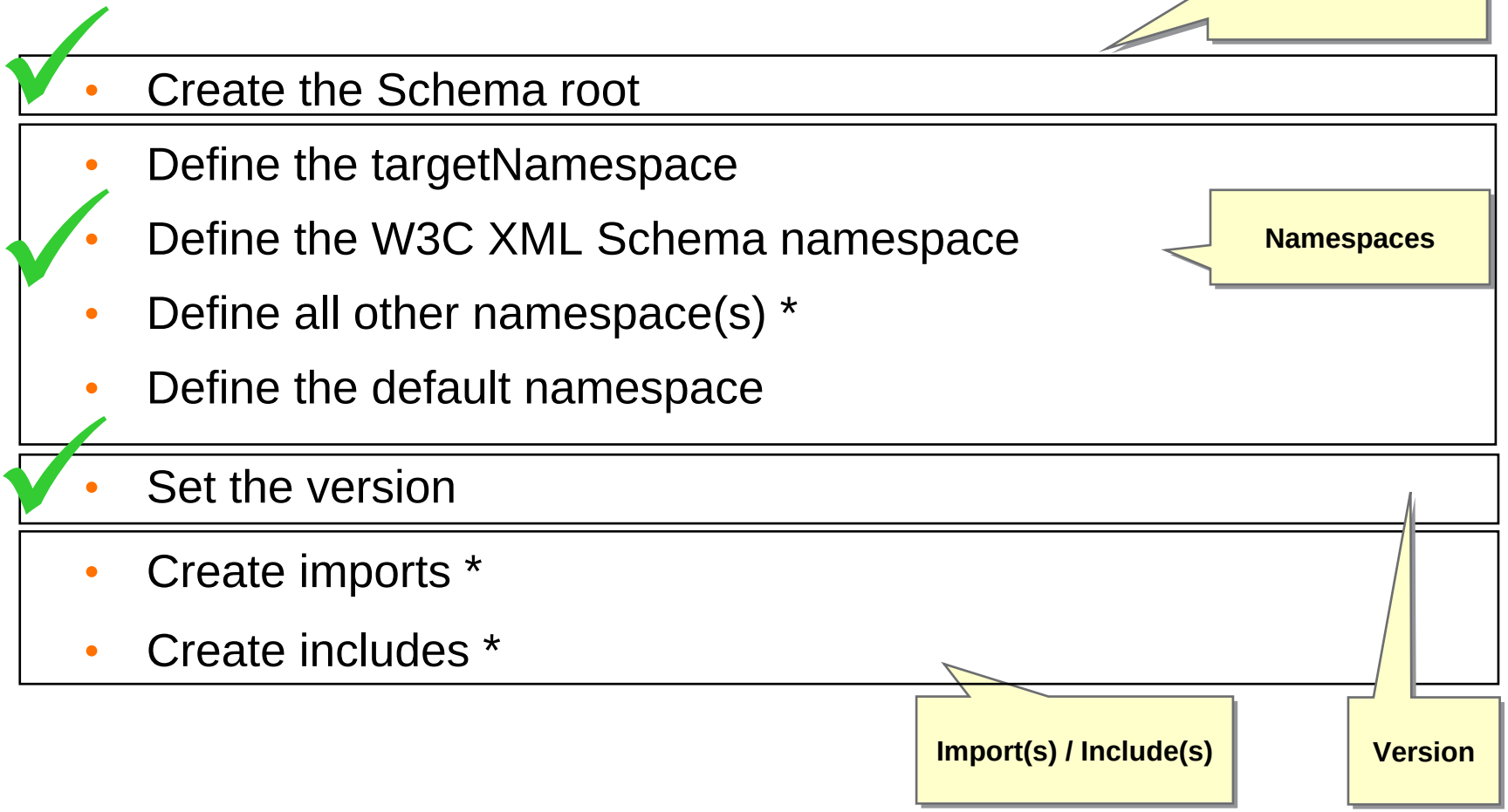
Document Header for: Inventory Availability Request

LMI



©Mark Crawford, LMI Government Consulting, April 2005

Create Document Schema header



©Mark Crawford, LMI Government Consulting, April 2005

Document Header for: Inventory Availability Request

LMI

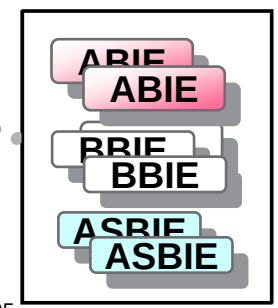
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:us:com:supplyinventory:inventorydepartment:1
  xmlns="urn:us:com:supplyinventory:inventorydepartment:1.0"
  xmlns:cac="urn:us:com:supplyinventory:enterprise:commonaggregatecomponents"
  xmlns:ccts="urn:un:unece:uncefact:documentation:corecomponenttechnicalspecificati
  on:2.0“
  elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0“ >
  <xsd:import
  namespace="urn:us:com:supplyinventory:enterprise:commonaggregatecompone  ts"
  schemaLocation="SI-Enterprise-CommonAggregateComponents.xsd"/>
  . . .
</xsd:schema>
```

Schema root

Namespaces

Import(s) / Include(s)

Version



it’s now time to populate the Schema body...

©Mark Crawford, LMI Government Consulting, April 2005

UBL Rules for XSD headers

[GXS1] UBL Schema MUST conform to the following physical layout as applicable:

XML Declaration

<!-- ====== Copyright Notice ====== -->

"Copyright © 2001-2004 The Organization for the Advancement of Structured Information Standards (OASIS). All rights reserved.

<!-- ====== xsd:schema Element With Namespaces Declarations ====== -->

xsd:schema element to include version attribute and namespace declarations in the following order:

- `xmlns:xsd`
- Target namespace
- Default namespace
- CommonAggregateComponents
- CommonBasicComponents
- CoreComponentTypes
- Unspecialised Datatypes
- Specialised Datatypes
- Identifier Schemes
- Code Lists

Attribute Declarations – elementFormDefault="qualified" attributeFormDefault="unqualified"

<!-- ====== Imports ====== -->

CommonAggregateComponents schema module

CommonBasicComponents schema module

Unspecialized Types schema module

Specialized Types schema module

Course materials follow this sequence

(as defined in the UBL NDR)

Document Header for: Inventory Availability Request

LMI

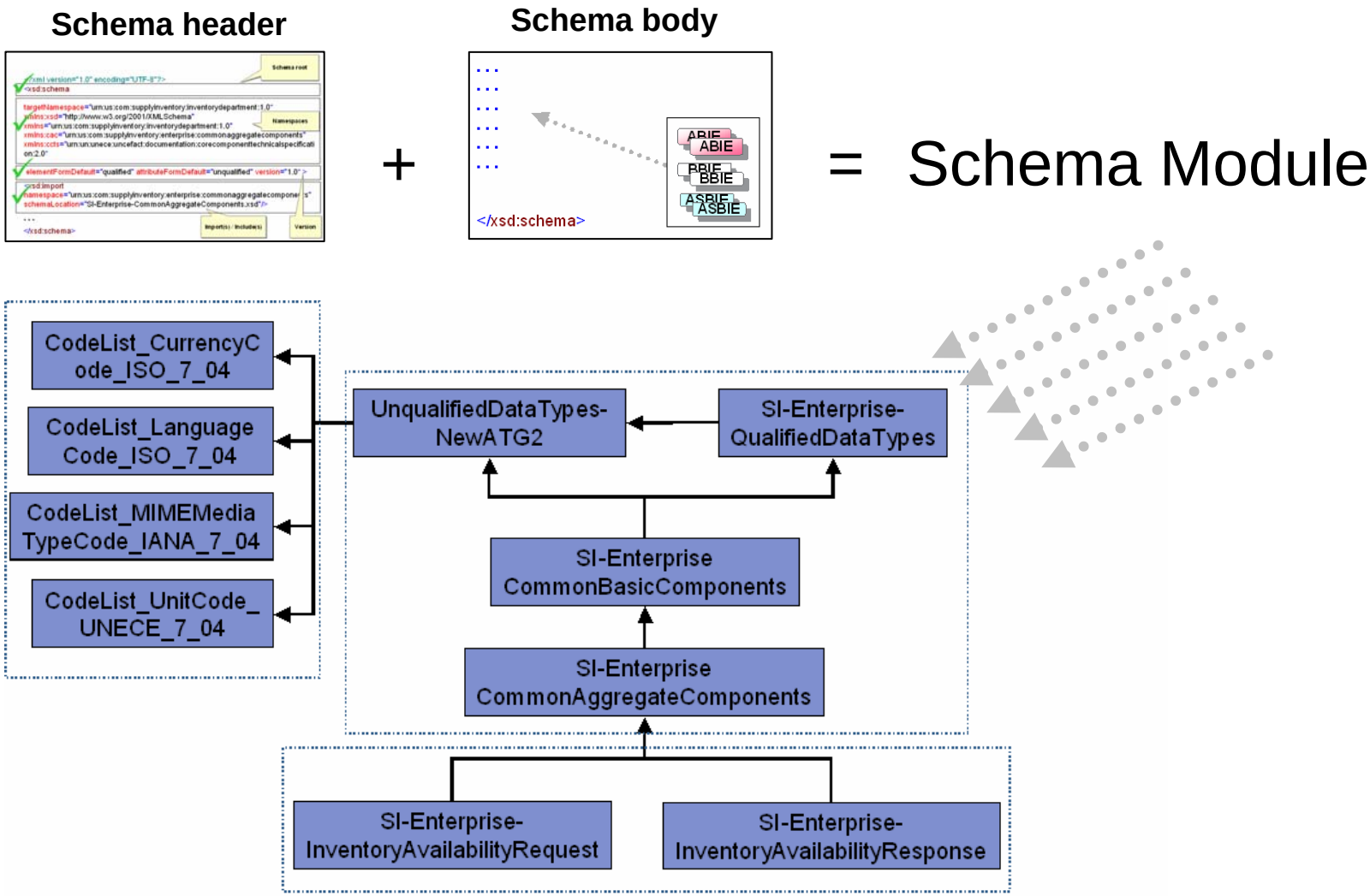
- One last word about Schema modularity
- The Schema headers are the 'control center' of Schema modularity

start of Schema root

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:us:com:supplyinventory:inventorydepartment:1.0"
xmlns="urn:us:com:supplyinventory:inventorydepartment:1.0"
xmlns:cac="urn:us:com:supplyinventory:enterprise:commonaggregatecomponents"
xmlns:ccts="urn:un:unece:uncefact:documentation:corecomponenttechnicalspecification:2.0"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0">

<xsd:import namespace="urn:us:com:supplyinventory:enterprise:commonaggregatecomponents"
schemaLocation="SI-Enterprise-CommonAggregateComponents.xsd"/>

. . .

</xsd:schema>
```

end of Schema root

Schema Modules (revisited)

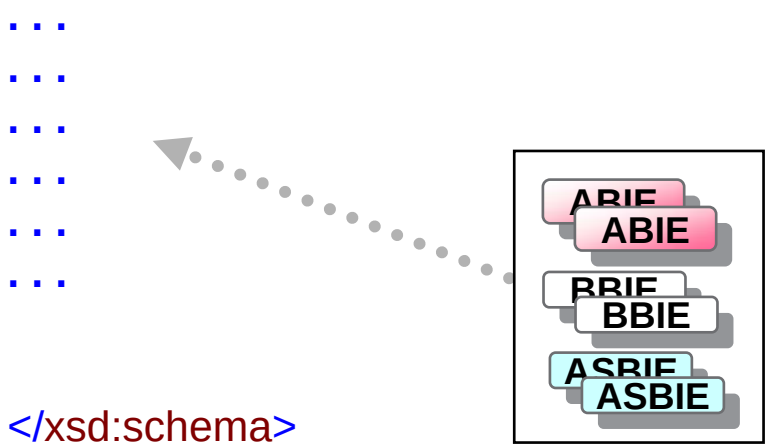


©Mark Crawford, LMI Government Consulting, April 2005

Populate Document Schemas

- Populate ‘Document’ Schemas
  - Create Message Assembly
  - Fill out ABIE / BBIE / ASBIE content
  - Create Aggregate Schema
  - Create Basic Schema
  - Define constraints (cardinality, optionality)
  - Create documentation

    (some documentation omitted from examples for brevity)



©Mark Crawford, LMI Government Consulting, April 2005

Populate Document Schemas

- Populate 'Document' Schemas
  - **Create Message Assembly**

Inventory Availability - Data Analysis Worksheet

XML Syntax		CCTS Dictionary Entry Name & ISO11179 data constructs						
XML Tag Name	**XML ComplexType Name**	**CCTS Dictionary Entry Name**	**ABIE/ ASBIE/**	**Object Class Qualifier**	**Object Class Term**	**Property Term**	**Property Term**	**Representation**
InventoryAvailability Request	**InventoryAvailabilityRequestType**	**InventoryAvailability_ Request. Details**	**ABIE**	**InventoryAvailability**	**Request**		**Details**	
InventoryOrganization	InventoryOrganizationType	InventoryAvailability_ Request. Inventory_ Organization. Inventory_ Organization	**ASBIE**	InventoryAvailability	Request	Inventory	Organization	
InventoryItem	InventoryItemType	InventoryAvailability_ Request. Inventory_ Item. Inventory_ Item	**ASBIE**	InventoryAvailability	Request	Inventory	Item	
SupplierOrganization	SupplierOrganizationType	InventoryAvailability_ Request. Supplier_ Organization. Supplier_ Organization	**ASBIE**	InventoryAvailability	Request	Supplier	Organization	
InventoryAvailability Response	**InventoryAvailabilityResponseType**	**InventoryAvailability_ Response. Details**	**ABIE**	**InventoryAvailability**	**Response**		**Details**	
SupplierOrganization	SupplierOrganizationType	InventoryAvailability_Response. Supplier_ Organization. Supplier_ Organization	**ASBIE**	InventoryAvailability	Response	Supplier	Organization	
InventoryItem	InventoryItemType	InventoryAvailability_Response. Inventory_ Item. Inventory_ Item	**ASBIE**	InventoryAvailability	Response	Inventory	Item	

ABIE

ABIE

remember these 2 ABIEs...

©Mark Crawford, LMI Government Consulting, April 2005

Populate Document Schemas

LMI

Populate 'Document' Schemas

- **Create Message Assembly**

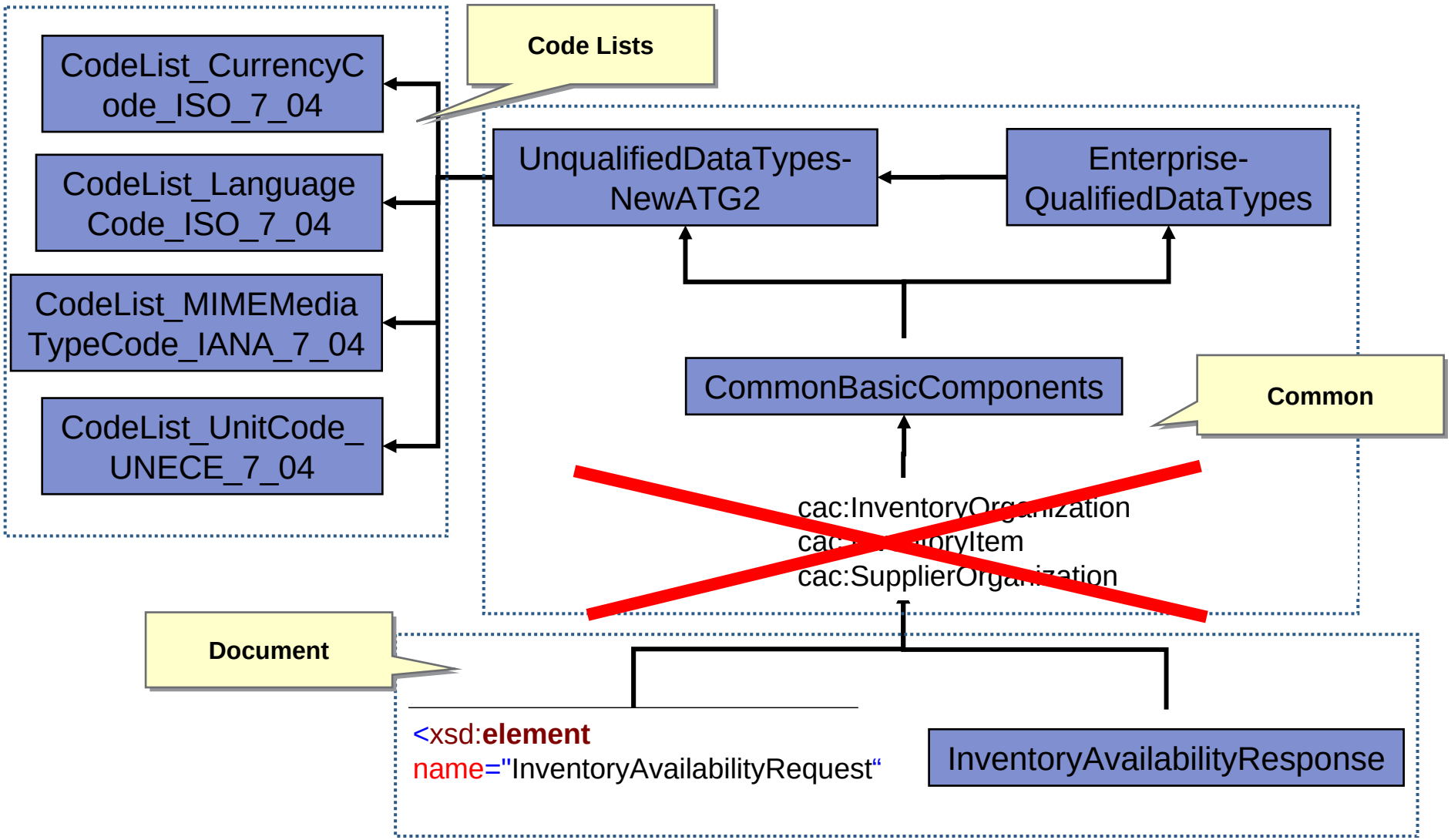
```
<xsd:schema ... > ...
    <xsd:element name="InventoryAvailabilityRequest" type="InventoryAvailabilityRequestType"/>
    <xsd:complexType name="InventoryAvailabilityRequestType">
        <xsd:annotation>
            <xsd:documentation>
                <ccts:CategoryCode>ABIE</ccts:CategoryCode>
                <ccts:DictionaryEntryName/>
                <ccts:Definition>This holds... </ccts:Definition>
                <ccts:ObjectClass/>
                <ccts:RepresentationTerm/>
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element ref="cac:InventoryOrganization"/>
            <xsd:element ref="cac:InventoryItem" maxOccurs="unbounded"/>
            <xsd:element ref="cac:SupplierOrganization"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

Root Element

ABIE

©Mark Crawford, LMI Government Consulting, April 2005

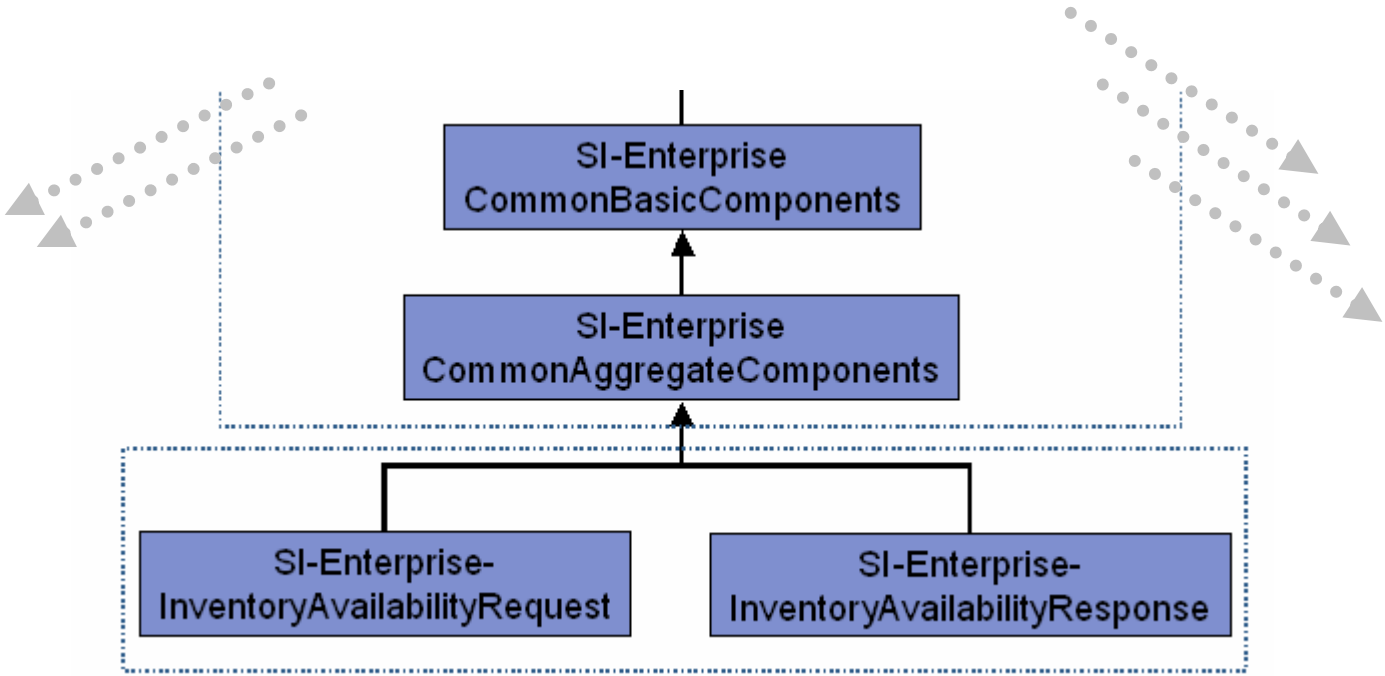
Populate Document Schemas



©Mark Crawford, LMI Government Consulting, April 2005

Document & Common Schemas

- Document Schemas ‘use’ the content defined in the common (or enterprise) Schemas
- Those Schemas will be used by any number of Document Schema modules



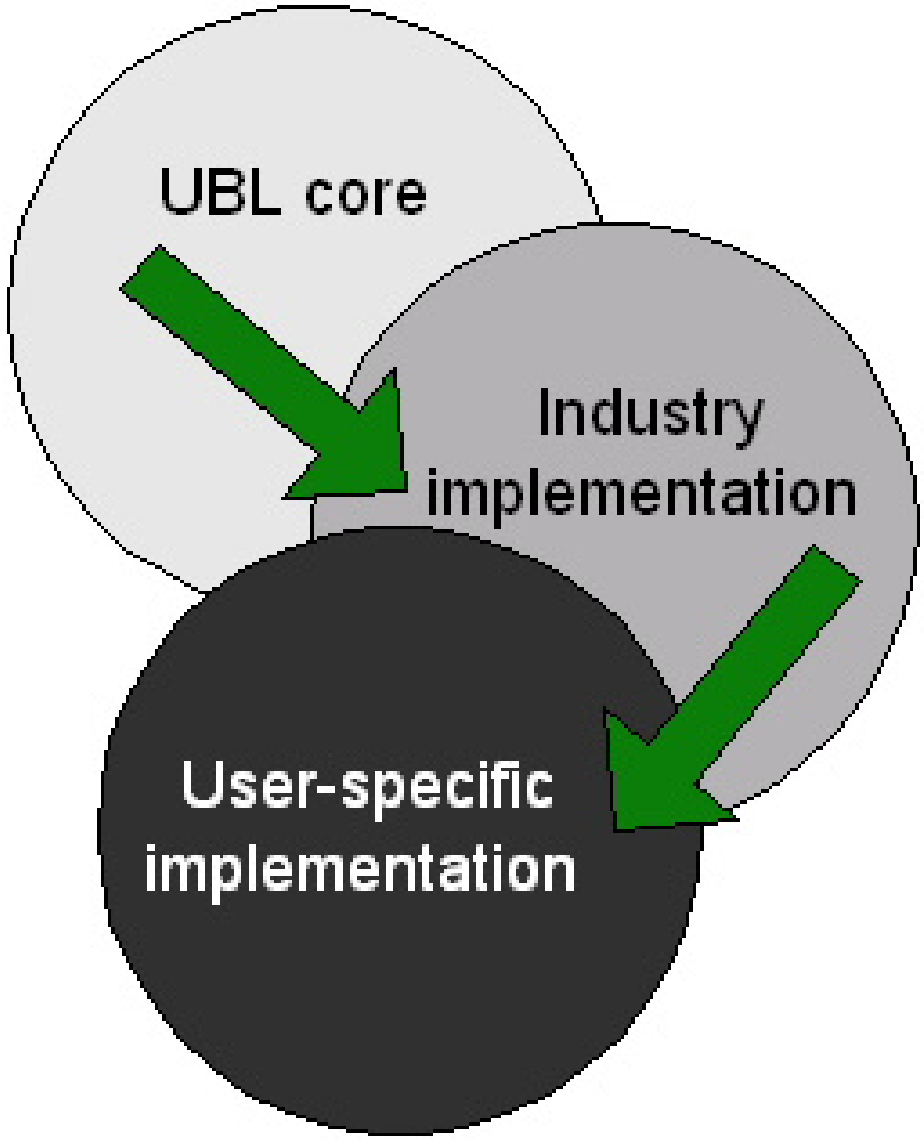
©Mark Crawford, LMI Government Consulting, April 2005

Customizing UBL

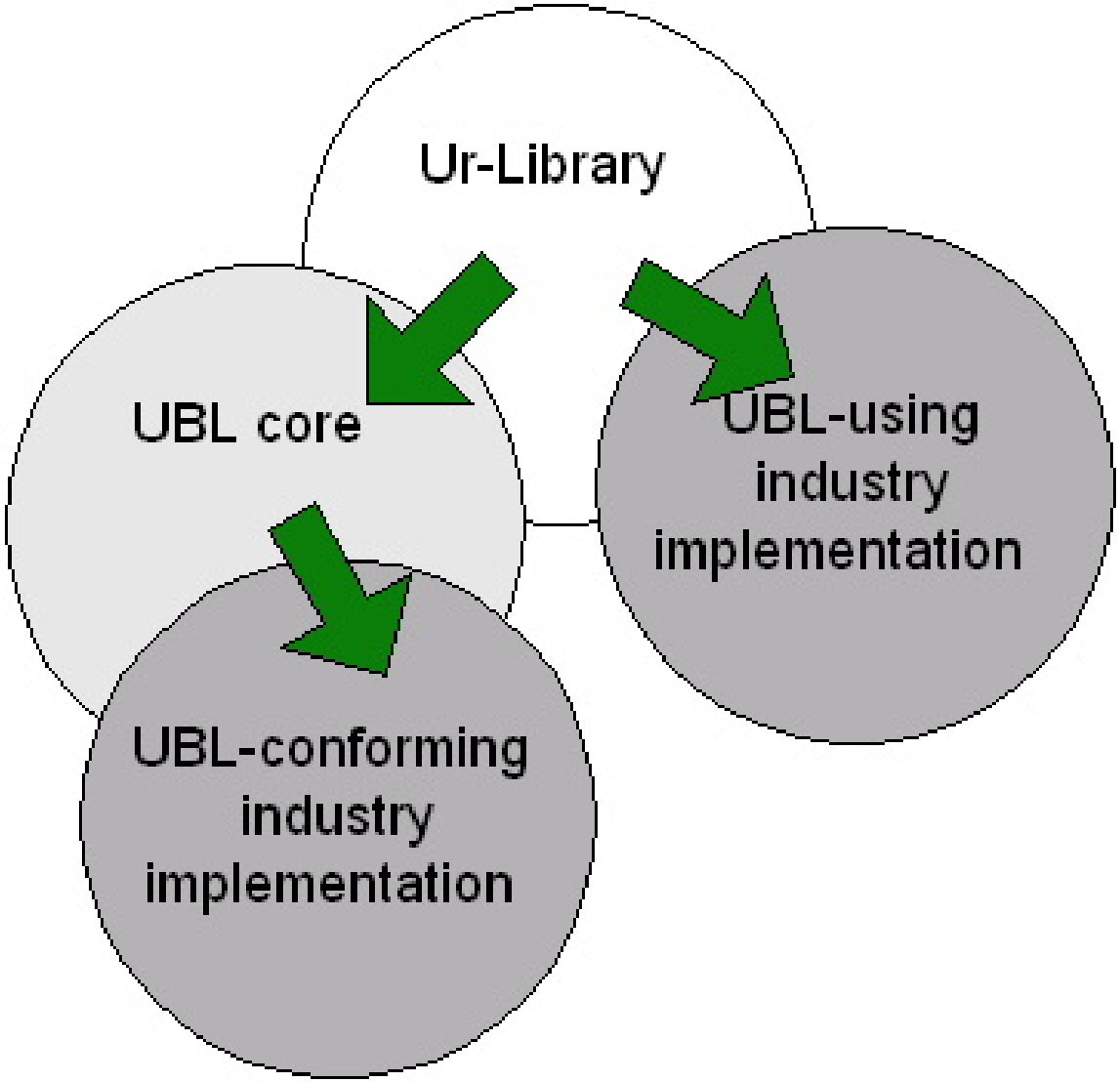
LMI

- Guidelines For The Customization of UBL v1.0
  - Customization will happen
  - It will be done by a wide range of users
  - Changes will be driven by real world needs
  - These needs will be expressed as context drivers

UBL Derivation - Conformant

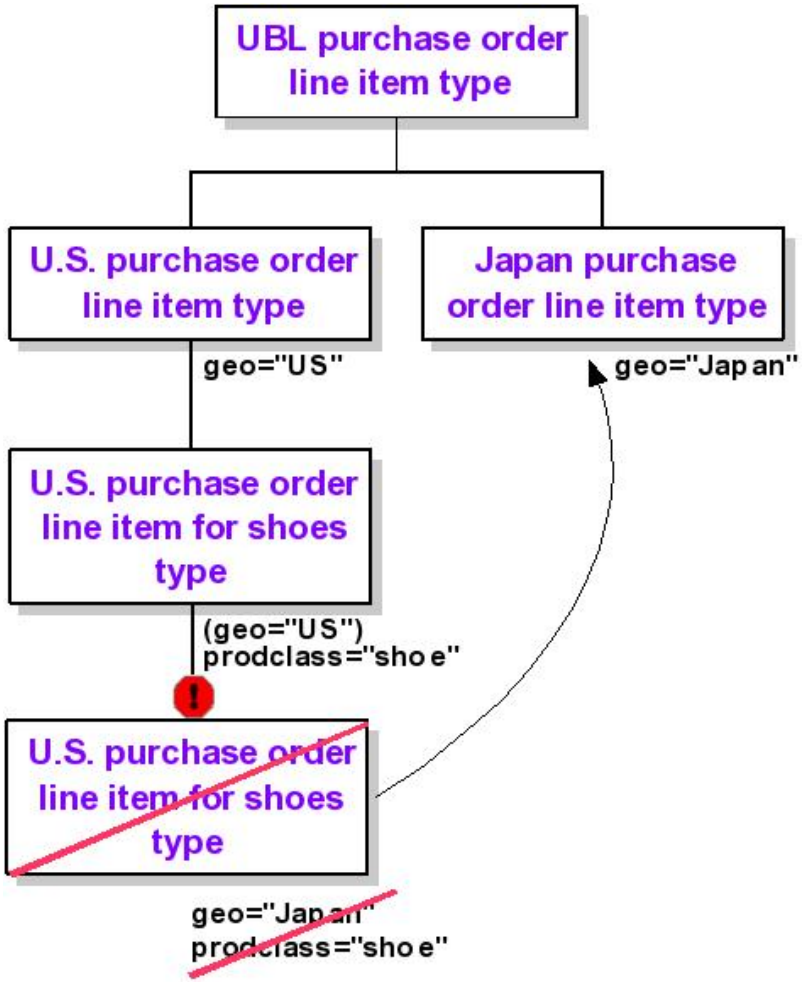


UBL Derivation – Conformant and Non-conformant



©Mark Crawford, LMI Government Consulting, April 2005

Limits on the Application of Context



Thank You