

# The Technical Framework for UN/CEFACT deliverables

General Discussion Presentation

# Outline for Discussions

- Requirements
- Issues

# **REQUIREMENTS**

# Background

- 2009 eBusiness Government Trade (eBGT) Gap Analysis
  - Stakeholder requirements?
  - What should be the deliverables?
  - Does CEFACT have the most effective Technical Architecture?
- 2010 Plenary accept findings and recommendations in principle, involving:
  - Alternative frameworks, procedures, structure
  - Forum and Bureau workshops debate issues
  - Plenary accept proposal to change - 'the way forward'.
- 2011 Principles put into practice
  - Bureau details new rules and organizational structures to ex-Forum
  - GME brainstorm (Washington)
  - Bureau seeks Plenary approval for new structures, rules and goals (July 2011)
- We have put much attention on the organization – now we need to address the technical framework.

# Product Architecture

- A business standard is composed of many parts. These may include:
  - The Business Requirements Specification (BRS)
  - The Requirements Specification Mapping (RSM)
  - The Core Components and Business Information Entities in the Core Component Library (CCL) that support the standard
  - The model of the document or documents to be exchanged in the business message
  - The eventual syntax solution or solutions
- Each of these deliverables are governed by one or more technical specifications.

# Syntax-neutral Specifications


- The Core Component Technical Specification
- The Data Type Catalogue
- UN/CEFACT Modeling Methodology
- The UN/CEFACT Context Methodology
- The UML Profile for Core Components
- Core Component Business Document Assembly
- UMM (templates for BRS and RSM)

# Syntax Specifications

- Naming and Design Rules for XML
- Message Design Rules (MDR) for EDIFACT
- CCTS to EDIFACT (needs to be verified and should include the possibility of mapping from the document model to EDIFACT)

# Issues

- Political
- Business
- Organizational
- Technical



In this presentation we  
are talking about this

# Gap Analysis - Messages

- There are a few common requirements (especially when we apply weighting to the responses from Government Agencies):
  - Invoice,
  - Order and
  - Customs Declaration
- There are gaps between these requirements and the perceived ability of UN/CEFACT to satisfy them.

# Gap Analysis – Syntax and Format

- UN/EDIFACT is still the dominant syntax and format currently used for eBusiness , Government and Trade;
  - remains a requirement for many stakeholders the future
  - more than UN/CEFACT XML
- There are four significant formats/syntaxes in use today;
  - UN/EDIFACT
  - OASIS UBL
  - In-house formats of XML
  - Adobe PDF

# Gap Analysis - Library Outreach

- Encourage rapid development of XML for CCTS.
- Work with the 7372 MA to more closely align the CCL with the TDED.
- Work to maintain backwards compatibility as the CCL is developed further to allow stable, mature business standards to be built.
- Create a Core Component Message Assembly specification to better enable messages to be developed from the CCL.
- Encourage greater efforts at convergence with other standards.

# Gap Analysis - Technical Framework

- There is little interest in the methodology specifications amongst the UN/CEFACT stakeholders or for this particular set of respondents.
  - A consequence of this could then be that the further development and maintenance of the methodology specifications should be moved to other organisations, such as ISO or OASIS.

# Gap Analysis - Future Work

- Library Improvement
- Conformance Paper
- Guidelines for Reference Data Models
- Possible "Electronic Trade Document Methodology"
  - Uses Common Reference Data Model
  - Consistently applied for paper, XML and UN/EDIFACT

→ Specification Evaluation Task Force – Sept. Forum

# Technical: General

- Packages – Concept Introduced in CCTS 3.0
  - Attempt to solve problem caused by Library expansion/complexity
  - Needs documenting
  - Sustained dissent amongst stakeholders and/or individuals
  - May be barrier to migration to 3.0
  - Should evaluate alternative approaches
- User guides needed for various specifications
  - Specifications have high entry level for understanding
  - Inconsistent understanding produces incompatible implementations.
- Versioning - Bringing into alignment the schema and business standard versions
  - Version wars between 2.01 and 3.0
  - Technical weakness in versions of XML always breaks backward compatibility
- Creating or adopting a scheme for generating globally unique identifiers
  - Manifestation of broader issue about policies for use of codes and identifiers
- Publication strategy
  - Policy driven – not market driven (6 monthly schedule)
  - Large maintenance burden created



Tim's thoughts

# Technical: General

- Define relationship between BRS, RSM, library, and end message (before syntax)
  - Missing a Document Architecture specification
- Product strategy document
  - Missing a high level view of architecture
  - CSDAT mark 2 (what should be - not documenting what we have)
- Code list maintenance and publication independent of syntax
  - Maintenance of content is fragile
  - Publication format geared to EDIFACT use
- Potentially abandon CCTS specifications as we know them
  - CCTS is not syntax-neutral (only works with XML), so why have an intermediary format?
    - May as well just create XML objects instead of Core Components, - save the effort and confusion
  - Indicates that 'vision' has been lost to 'process'

# Technical: XML

- Packages
  - May not be taking best advantage of XML architecture
- Code lists encoded into schema
  - This creates restrictions to using updated code values
- Support for more than one version of CCTS
  - 2 different NDRs makes implementations incompatible
- Relative path names
  - Not best practice (subject to failure)
- Should the version number be part of the schema name?
  - Problematic with frequent versioning.

# Technical: EDIFACT

- Look at CCTS 3.0 in relation to this
  - Aligning EDIFACT, TDED and CCL should be a goal
- Get implementation verifications
  - Can we learn from how users implement EDIFACT

# Technical: CCTS

- Context not well defined
  - UCM has become disconnected from CCTS methodology
- Backwards compatibility – what is the definition of this in terms of CCTS, probably a matter of the DTC
  - A challenge with being syntax-neutral
  - Requires good management – eg ‘core’ plus ‘context’
- Properties as distinct entities
  - Appears to over-engineer specification
  - What is the business value?
- Allow prepositions in DENs
  - The CCL needs to inform the CCTS of best practice – not the other way around.
- The change in data types from unqualified and qualified to core and business
  - Need some clarity and stability in this area
- Business value domains
  - Needs relating to UCM
- Potentially remove associations (ASCCs) from the main specification
  - CCs should context free, but associations create context.
  - Basic principles need reviewing after experience in implementation.
- What is the role of the CC library with regard to existing standards?
  - Issue suggests a deeper problem – how do others use this stuff in practice?
- Better structure the document to distinguish the structural versus maintenance functions of CCTS
  - Currently its scope is confused
  - It is a weak specification

# Technical: Data Types

- Code lists bound into data types – potentially drop this
- Number/Numeric
- Indicator – should be more than just Boolean values
- Codes versus Identifiers
- Hierarchical data types – allow certain data types to be subtypes of others, and restriction from the BCC to the BBIE can change the data type from a primary to a subtype
- Unusual data types (e.g. some of the items in UNCL 2379)
- Cover 8601 basic date values
  
- Suggests that neither version has addressed its user requirements
- Change requests already

# Technical: Others

- **UCM**
  - Specific mechanisms for message/document creation not fully defined
  - Need better guidance on how to produce the context graph
  - Need some simplification of the concepts
  - Lack of understanding in this whole area, why?
  - May be a technical solution to an ill defined requirement
- **UPCC**
  - Terminology regarding packages and libraries needs to be reconciled with CCTS
  - Lack of common vision and coordination
- **CCBDA**
  - Incorporate packaging concept?
  - Add the ability of defining message-specific associations and ordering
  - Needs to be part of an overall document architecture approach
  - Example of fragmented activity creating a solution then retrofitting to other specifications

# Summary of Situation

- Challenged by maintenance of artifacts
- Challenged by creating technical specifications
- Challenged by using technical specifications
- Challenged by creating XML
- Challenged by maintaining XML
  
- At risk with EDIFACT maintenance

# What do we do about this?

- Agree on what the 'end game' should be:
  - What an effective framework for UN/CEFACT deliverables would look like.
- Plot what we have now
- Plan how to reach the 'end game'
  - Who does what
  - Stop doing things that don't fit this plan
- Manage the completion of these projects
  - Get the right resources
- Manage expectations
  - Communicate value