Address Levels

Business Use Alignment
Introduction

• Objective is to provide layers of address granularity tailored to business use

• Address use levels
  – Level 0 = handwritten postal address – machine parsed
  – Level 1 = in country simple postal address – legacy
  – Level 2 = extended postal address – advanced features
  – Level 3 = shipping / delivery address; large organization
  – Level 4 = facilities management; universal / exotic / global

• Share common noun definitions

• Share validation rules and quality factors

• Provide means to manage the quality of address content

• Provide global language and format support
## Delivery Quality Measurement

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Multiple lines of unspecified address information</td>
</tr>
<tr>
<td>1</td>
<td>Identified lines of address information</td>
</tr>
<tr>
<td>2</td>
<td>Partial element identification (City-State-ZIP)</td>
</tr>
<tr>
<td>3</td>
<td>Full element identification</td>
</tr>
<tr>
<td>4</td>
<td>Full element identification with some validation (valid State)</td>
</tr>
<tr>
<td>5</td>
<td>Full element identification with some validation and cross validation</td>
</tr>
<tr>
<td>6</td>
<td>Matching address to Postal Database (CASS certified address matching software used)</td>
</tr>
<tr>
<td>7</td>
<td>Complete and correct address (Delivery Point Validation)</td>
</tr>
<tr>
<td>8</td>
<td>Know recipient has not reported a move (Move Update)</td>
</tr>
<tr>
<td>9</td>
<td>Know delivered to person / entity</td>
</tr>
<tr>
<td>10</td>
<td>Know the delivery was made to the right person / entity</td>
</tr>
</tbody>
</table>
Technology

- Using W3C Schema to provide layers of increasingly refining definitions based on business use
- Use OAGIS V8 methods to restrict syntax to best-practice techniques
- Enable use of ebXML AssemblyDoc technology
- Provide migration from legacy address formats
- Harmonization of existing and emerging standards to single common base noun dictionary and use templates
Syntax Objective

- Create single schema definition that is capable of support all four levels of use with increasing detail granularity
- Support and harmonize noun dictionaries across multiple legacy implementations by allowing substitution underneath parents
- Use of ebXML UID technique to equate like elements in dictionary
Technical Detail

• Address
  – Address Line

• Line type
  – Building
    » Number, Street Name, Street Type
  – PO Box
    » Number, Type
  – Business Location
    » Suite, Number, Street Name, Street Type

Schema Overlay Definitions
XML Methods

• Two methods:
  – Derivation by extension: allows type B to extend type A without touching type A and without repeating the elements in type A
  – Replacement: substitute in another different structure within the hierarchy with the same parent name, but different elements

• We need to use both to achieve syntax goals
Schema limitations

- Schema can show you the complete set of all possible structure derivatives
  - A, (B | C), D, E, (F | G)
- Change denominated lists by restriction only
- Cannot tell you if you use C you must use G, i.e context rules and business rules
- Therefore - need to use ebXML AssemblyDoc templates
ebXML AssemblyDocs

• Consist of four parts
  – Assembly Structure(s)
  – Business Context Rules
  – Content type references
  – Data Validation rules

• Specification available end of May