CABINET OFFICE

Office of the *e-Envoy*

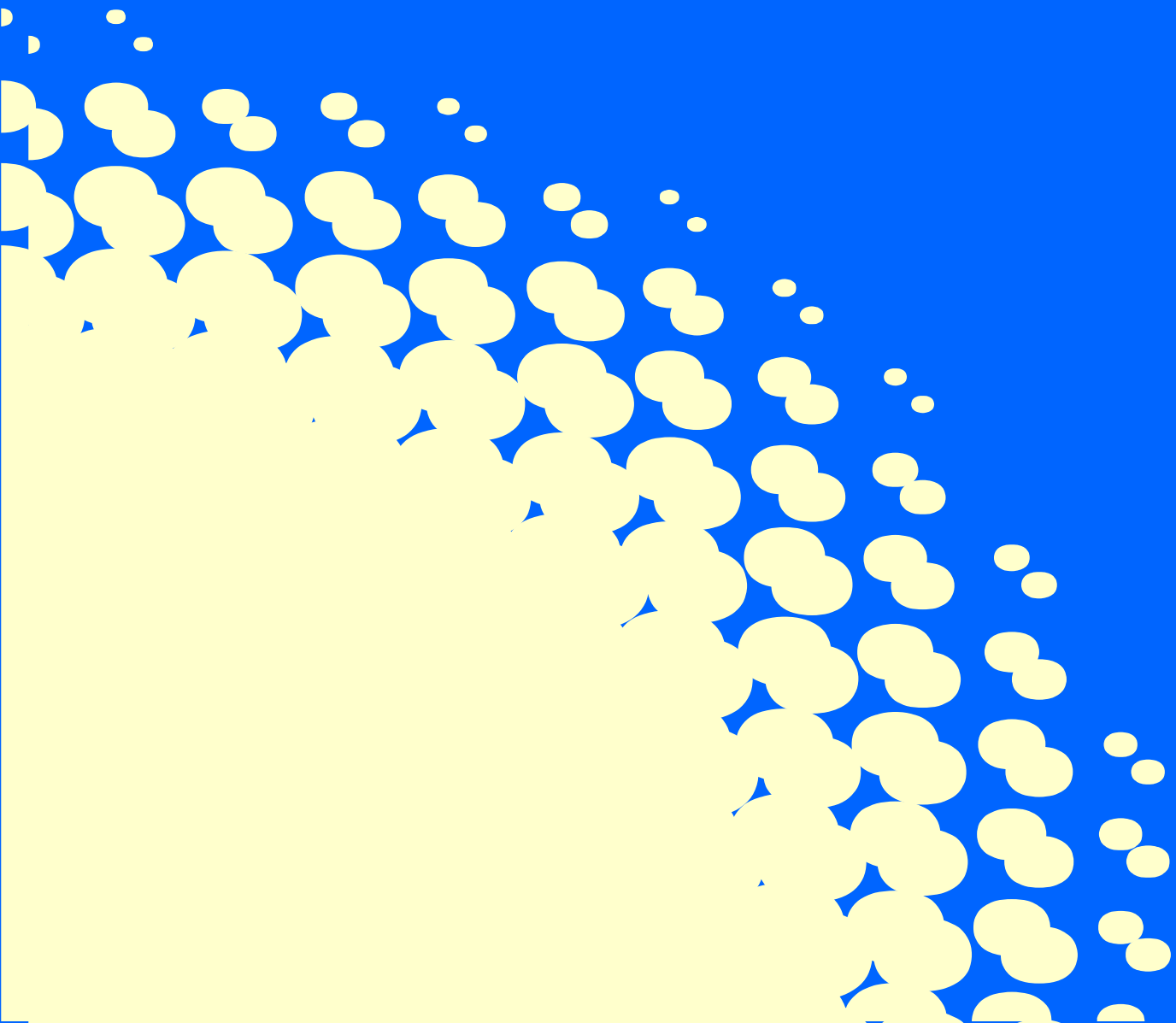
Leading the drive to get the UK online

delivering UK online

Proposal: Codelists

1.0

April 2003



Document Control

Abstract

In many cases, there is a need to use sets of pre-defined codes (such as country and currency codes) to help achieve interoperability between XML-based systems. XML documents using these codes require both a mechanism to check that valid codes are being used and a mechanism to look up the meanings of codes. The intention is to represent these code sets as XML documents.

Although this paper considers the requirements of UK Government for the representation of code sets, the requirements are thought to be sufficiently general to be adopted elsewhere.

A solution is proposed that is directly compatible with the code list format being proposed by the OASIS UBL Technical Committee, with extensions to allow the looking up of code meanings.

Current Version

Date	Version	Status	Editor	Comment
25 April 2003	1.0	release	Paul Spencer paul.spencer@boynings.co.uk	Re-presented as a proposal for international adoption. Only the UBL-compatible mechanism included.

Change History

Date	Version	Status	Editor/ Author	Comment
1 April 2003	1.0b	draft	Paul Spencer paul.spencer@boynings.co.uk	UBL-compatible mechanism added using the latest version of UBL material. The "optimized" mechanism altered to be compatible with the latest UBL version. Title changed to "Discussion Paper" since there are now two options.
16 Dec 2002	1.0a	draft	Paul Spencer paul.spencer@boynings.co.uk	First draft

Executive Summary

In many cases, there is a need to use sets of pre-defined codes (such as country and currency codes) to help achieve interoperability between XML-based systems. XML documents using these codes require both a mechanism to check that valid codes are being used and a mechanism to look up the meanings of codes. The intention is to represent these code sets as XML documents.

The major requirements for a mechanism for the representation of code sets are:

- it must support long-term archiving of documents;
- it must support versioning;
- it must support validation of instance documents;
- it must be possible to look up the meanings of codes ("dereferencing");
- it must support multiple definitions of codes
- it must support multi-lingual definitions of codes;
- it must be compatible with UBL Code List XML Schema files; and
- it must be possible to carry both a code and (optional) value in the message.

Also, where possible, the representation mechanism should be based on existing standards.

This proposal presents a solution that meets all the requirements above. It is directly compatible with the UBL schema format, using `xs:appinfo` elements to hold the additional information required for dereferencing.

It is proposed that the OASIS UBL TC and UN/CEFACT review this proposal with a view to adopting it either unchanged or modified to meet a revised set of requirements.

Contents

Introduction

In many cases, there is a need to use sets of pre-defined codes (such as country and currency codes) to help achieve interoperability between XML-based systems. XML documents using these codes require both a mechanism to check that valid codes are being used and a mechanism to look up the meanings of codes. The intention is to represent these code sets as XML documents.

This paper considers a set of requirements for the representation of code sets and proposes a solution. In general, these code sets will be international in nature (for example, ISO standards) so an internationally agreed mechanism is preferred. Although the requirements are based on those of the UK Government, they are thought to be sufficiently general that the proposed solution meets international needs.

In this document, the term "code set" is used to indicate a set of codes, and possibly additional information about the codes, while the term "code list" is used to indicate an XML representation of a code set.

Requirements

Mandatory

The following requirements must be met for a code list mechanism to be useful.

Support long-term archiving of documents

Many documents require long-term archiving. To achieve this, the documents must be able to indicate the version of a code set that is in use. They should do this both in the code list reference in the instance document (so the correct set is accessed) and in any document metadata (so that the code list(s) required can be archived with the document).

Versioning

A code list must indicate its version number. It may also provide dates between which it is valid. A reference to a code list must indicate the version to which it refers.

Validation

It must be possible to discover whether a code in use is a valid member of the indicated code set.

Looking up Code Meanings

It must be possible to find the meaning of any code in the code set.

Multiple definitions (e.g. short and long)

The mechanism used for holding code sets must support multiple definitions. Specifically, it must be possible to hold both a short description, a long description and a symbol (with information on the correct positioning of the symbol relative to the value).

Multi-lingual

The mechanism used for holding code sets must support extracting information in multiple languages.

Must be compatible with UBL Code List XML Schema files

It is a requirement is that it must be possible to validate that codes used in a document are valid. A mechanism to achieve this could be to have direct compatibility with, or the ability to generate, W3C XML schemas to the format described in the OASIS white paper "UBL Code List Rules: A White Paper" [1].

Must be able to carry both a code and (optional) value in the message

A requirement in the health service (and possibly elsewhere) is that both a code and a meaning can be carried in a document. As an example of the use of this, if a person selects a value from a drop-down list on a web page, the text selected as well as a code describing it must be transmitted to health systems for audit purposes, even though the receiving application might ignore the text and use the indicated code list to look up a meaning.

Desirable

Use Existing Specifications

Where possible, existing specifications should be used. In particular, this proposal will become an input for consideration by the OASIS e-Government Technical Committee, so use of existing OASIS specifications is beneficial.

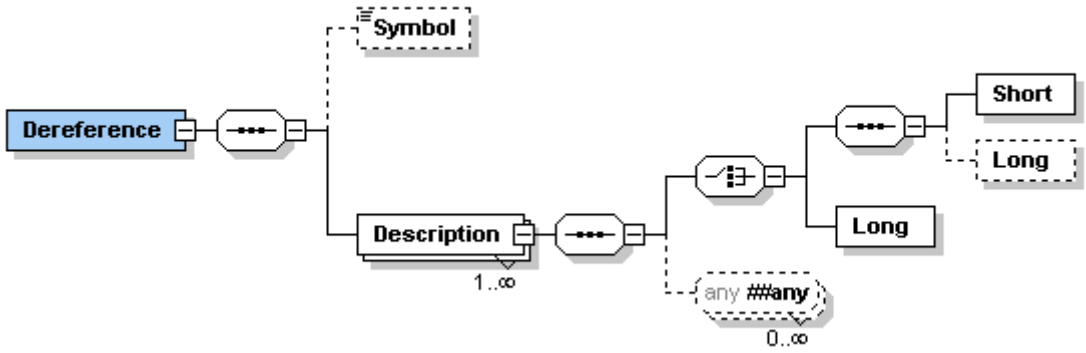
Code List Format

Proposed UBL-Compatible Code List Format

The intention with this format is to allow use of the UBL code list format unaltered, while adding dereferencing facilities through the use of `xs:appinfo` elements. Note that the current UBL format puts some of this information into `xs:documentation` elements to create XHTML documentation. This can be left, or the documentation can be derived from the more structured format shown here.

The Code List Format

The diagram shows the element structure of a code list. The `Dereference` element will be a child of an `xs:appinfo` element describing each enumerated value allowed in the code list (see the example below)



An `xs:any` element is included to allow additional information, but this could be removed if preferred.

The following attributes are used:

Element: Symbol

Name	Type	Use	Allowed Values
position	xs:token	optional	before-value after-value
spaceBetween	xs:boolean	optional	true false

Element: Description

Name	Type	Use	Example
xml:lang	xs:language	required	en

The following listing shows an extract from a currency code list. The `xs:documentation` elements from the UBL original have been left in except in the `xs:enumeration` elements, where they are replaced by the `xs:appinfo` elements.

The simple data type `CodeContentType` contains the enumerated values contained in the code set with the dereferencing information in `xs:appinfo` elements.

The complex data type `CodeType` is the type that should be applied to an element that represents a coded value. This has the following attributes:

Name	Type	Fixed Values?	Example
listID	xs:token	yes	ISO 4127
listAgencyID	xs:token	yes	6
listVersionID	xs:string	yes	0.3
href	xs:anyURI	yes	http://example.com/4127-UBLcompatible.xsd
description	xs:string	no	US Dollar

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

targetNamespace="http://example.com/iso4127ExtendedCurrencyCodeSample"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:code="http://example.com/codelists"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  xmlns="http://example.com/iso4127ExtendedCurrencyCodeSample">

  <xs:simpleType name="CodeContentType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:div class="Core_Component_Type">
          <xhtml:p>Code. Type</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Identifier">
          <xhtml:p>ISO 4127</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Agency._Identifier">
          <xhtml:p>6</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Version._Identifier">
          <xhtml:p>0.3</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Valid._From">
          <xhtml:p>2002-01-01</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Valid._To">
          <xhtml:p>2004-12-31</xhtml:p>
        </xhtml:div>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="USD">
```

```
        <xs:annotation>
          <xs:appinfo>
            <code:Dereference>
              <code:Symbol position="before-value"
spaceBetween="false">$</code:Symbol>
              <code:Description xml:lang="en">
                <code:Short>US Dollar</code:Short>
              </code:Description>
            </code:Dereference>
          </xs:appinfo>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="GBP">
        <xs:annotation>
          <xs:appinfo>
            <code:Dereference>
              <code:Symbol position="before-value"
spaceBetween="false">&#x00a3;</code:Symbol>
              <code:Description xml:lang="en">
                <code:Short>GB Pound</code:Short>
              </code:Description>
            </code:Dereference>
          </xs:appinfo>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="JPY">
        <xs:annotation>
          <xs:appinfo>
            <code:Dereference>
              <code:Symbol position="before-value"
spaceBetween="false">&#x00a5;</code:Symbol>
              <code:Description xml:lang="en">
                <code:Short>Yen</code:Short>
              </code:Description>
            </code:Dereference>
          </xs:appinfo>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="EUR">
        <xs:annotation>
          <xs:appinfo>
            <code:Dereference>
              <code:Symbol position="after-value"
spaceBetween="false">&#x20ac;</code:Symbol>
              <code:Description xml:lang="en">
                <code:Short>Euro</code:Short>
              </code:Description>
            </code:Dereference>
          </xs:appinfo>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="CodeType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:div class="Core_Component_Type">
          <xhtml:p>Code. Type</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Identifier">
          <xhtml:p>ISO 4127</xhtml:p>
        </xhtml:div>
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Agency._Identifier">
          <xhtml:p>6</xhtml:p>
        </xhtml:div>
```

```
      </xs:documentation>
      <xs:documentation>
        <xhtml:div class="Code_List._Version._Identifier">
          <xhtml:p>0.3</xhtml:p>
        </xhtml:div>
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="CodeContentType">
        <xs:attribute name="listID" type="xs:token" fixed="ISO 4127"/>
        <xs:attribute name="listAgencyID" type="xs:token" fixed="6"/>
        <xs:attribute
          name="listVersionID" type="xs:string" fixed="0.3"/>
        <xs:attribute
          name="href" type="xs:anyURI"
          fixed="http://example.com/4127-UBLcompatible.xsd"/>
        <xs:attribute name="description" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Format of a Reference

A format for a reference to this code list is:

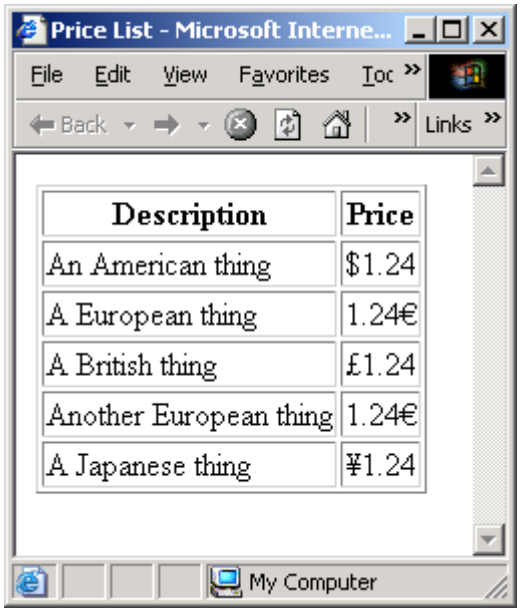
```
<?xml version="1.0" encoding="UTF-8"?>
<Prices
  xmlns:code="http://example.com/iso4127CurrencyCodeSample"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="4127-test-optimized.xsd">
  <Item>
    <Description>An American thing</Description>
    <Price>
      <Amount>1.24</Amount>
      <Currency listID="ISO 4127" listAgencyID="6" listVersionID="0.3"
href="4127-listID-optimized.xml">USD</Currency>
    </Price>
  </Item>
  <Item>
    <Description>A European thing</Description>
    <Price>
      <Amount>1.24</Amount>
      <Currency listID="ISO 4127" listAgencyID="6" listVersionID="0.3"
href="4127-listID-optimized.xml">EUR</Currency>
    </Price>
  </Item>
  <Item>
    <Description>A British thing</Description>
    <Price>
      <Amount>1.24</Amount>
      <Currency listID="ISO 4127" listAgencyID="6" listVersionID="0.3"
href="4127-listID-optimized.xml">GBP</Currency>
    </Price>
  </Item>
  <Item>
    <Description>Another European thing</Description>
    <Price>
      <Amount>1.24</Amount>
      <Currency listID="ISO 4127" listAgencyID="6" listVersionID="0.3"
href="4127-listID-optimized.xml">EUR</Currency>
    </Price>
  </Item>
  <Item>
    <Description>A Japanese thing</Description>
    <Price>
      <Amount>1.24</Amount>
      <Currency listID="ISO 4127" listAgencyID="6" listVersionID="0.3"
href="4127-listID-optimized.xml">JPY</Currency>
```

```
    </Price>
  </Item>
</Prices>
```

This shows all the attributes required on a `Currency` element, but omits the optional `description`. However, the UBL schema style creates these as attributes with "fixed" values, so they can all be omitted. Whether this is done or not depends on the degree to which the document format is to be tied to the W3C XML Schema syntax. By explicitly including the values, other schema languages can be used in the future without the need to change the instance documents.

A Stylesheet for Dereferencing

An XSLT stylesheet can be applied to this to combine the information in the XML document with information in the code list. The following display was created using the stylesheet shown below:




```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:code="http://example.com/codelists"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xsl:template match="Prices">
    <html>
      <head>
        <title>Price List</title>
      </head>
      <body>
        <table border="1">
          <tr>
            <th>Description</th>
            <th>Price</th>
          </tr>
          <xsl:apply-templates select="Item"/>
        </table>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="Item">
```

```
    <tr>
      <td>
        <xsl:value-of select="Description"/>
      </td>
      <td>
        <xsl:apply-templates select="Price" mode="coded"/>
      </td>
    </tr>
  </xsl:template>
  <xsl:template match="*" mode="coded">
    <xsl:variable name="code" select="*[@listID]"/>
    <xsl:variable name="codelist" select="document(*[@listID]/@href)"/>
    <xsl:variable name="item"
select="$codelist/xs:schema/xs:simpleType[@name='CodeContentType']/xs:r
estriction/xs:enumeration[@value=$code]"/>
    <xsl:choose>
      <xsl:when test="$item//code:Symbol[@position='before-value']">
        <xsl:value-of select="$item//code:Symbol"/>
        <xsl:if test="$item//code:Symbol[@spaceBetween='true']">
          <xsl:text> </xsl:text>
        </xsl:if>
        <xsl:value-of select="*[not(@listID)]"/>
      </xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="*[not(@listID)]"/>
        <xsl:if test="$item//code:Symbol[@spaceBetween='true']">
          <xsl:text> </xsl:text>
        </xsl:if>
        <xsl:value-of select="$item//code:Symbol"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

The template `<xsl:template match="*" mode="coded">` provides the dereferencing. In this case, we are looking up a symbol, which contains information relating to it positioning, and so is the most complex form of dereferencing. Simpler forms (for example, translating a country code to a country text string) can easily be achieved in two lines of XSLT.

The UBL format as currently presented in the white paper [1] does not meet the validation requirements of the UK Government. This requires the addition of an optional `description` that can be carried with a coded value. Whilst this could be held as an additional element, it is preferred that it be carried as an attribute of the element that contains the coded value.

To use this dereferencing mechanism, a further `href` attribute is required to be carried with the coded value to identify the code list to be used for dereferencing.

Current Usage

The proposed format is currently in use in a UK Government Project, and has been found to work successfully. Each code list in use has been generated from existing UN/CEFACT lists using a simple XSLT stylesheet. This produces a schema with the correct enumeration values and whatever information is available from the existing UN/CEFACT XML documents.

The current work could be made available for inspection if required.

Proposal

It is proposed that the OASIS UBL TC and UN/CEFACT review this proposal with a view to adopting it either unchanged or modified to meet a revised set of requirements.

Since the UBL TC and UN/CEFACT are now working together on code lists, it is hoped that UN/CEFACT will maintain lists incorporating many of the major world languages, while individual governments could maintain lists for other languages if required.

References

1. UBL Code List Rules: A White Paper (note that the paper was not at the advertised location at the time of writing) http://www.oasis-open.org/committees/ubl/ndrsc/archive/wp-ubl-codelist-01

Online copies of this document will be made available at: www.e-envoy.gov.uk

Office of the e-Envoy, Stockley House, 130 Wilton Road, London, SW1V 1LQ