EBU – TECH 3336

EBU Reference Data & Classification Schemes

Source: ECM

Status: Specification v.1.1

Geneva
September 2011

[*] Page intentionally left blank. This document is paginated for recto-verso printing

Introduction

The EBU has developed several Technical Specifications on Metadata e.g. to support business to business exchange of content, or more specifically exchange news or archive material. Each of these specifications relies to varying extent on the use of common reference vocabularies for interoperability purposes.

Reference data can be of different nature such as technical information like different picture formats, or descriptive metadata such as content genres.

The present document defines the metadata schema used by EBU to publish reference data in the form of xml classification schemes.

The EBU has identified several sets of reference data available as web resources.

Each set of reference data is provided as a default reference but can be replaced entirely by another similar set of reference data or customised to cover particular user needs.

The EBU schema for Classification Scheme further allows defining mapping to other Classification Schemes.

This information can be represented in alternative formats such as SKOS.

[*] Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for recto-verso printing

EBU Reference Data & Classification Schemes

EBU Committee	*First Issued*	*Revised*	*Re-issued*
ECM	July 2009		

Keywords: Metadata, Schema, Dublin Core, P-META, Tech 3293, Radio, Television, Reference Data, Classification Scheme, Vocabulary, Genre, SKOS

1. Scope

The current specification aims first at defining the information needed to describe and maintain hierarchical reference vocabularies.

In order to be used and processed by machines, such vocabularies need to be represented in an IT friendly format. The representation format used in this specification is XML.

Vocabularies can also be represented in alternatives formats, such as SKOS, for which mapping will be provided.

2. XML representation and definitions

2.1 Introduction

A reference vocabulary is a hierarchical collection of terms and related sub-terms.

NOTE: the use of a hierarchical collection of terms and sub-terms has been preferred to a flat list of terms as it facilitates the xml transformation into other formats such as SKOS.

One term in one vocabulary can have its equivalent (exact, narrow or broad match), so called 'mapping term', in another vocabulary.

XML has been chosen as the representation format as it is widely used (e.g. MPEG, DVB, TV-Anytime) and facilitates the transformation of data in various formats, if needed.

2.2 Namespace

The namespaces are in compliance with the namespace conventions defined in Tech 3295 (P_META) and RFC 174 (EBU namespace).

Schema namespace:

- urn:ebu:metadata-schema:SchemaName_YYYYMMDD (Month and Day are optional)

Example: urn:ebu:metadata-schema:EBU_CS_20090701

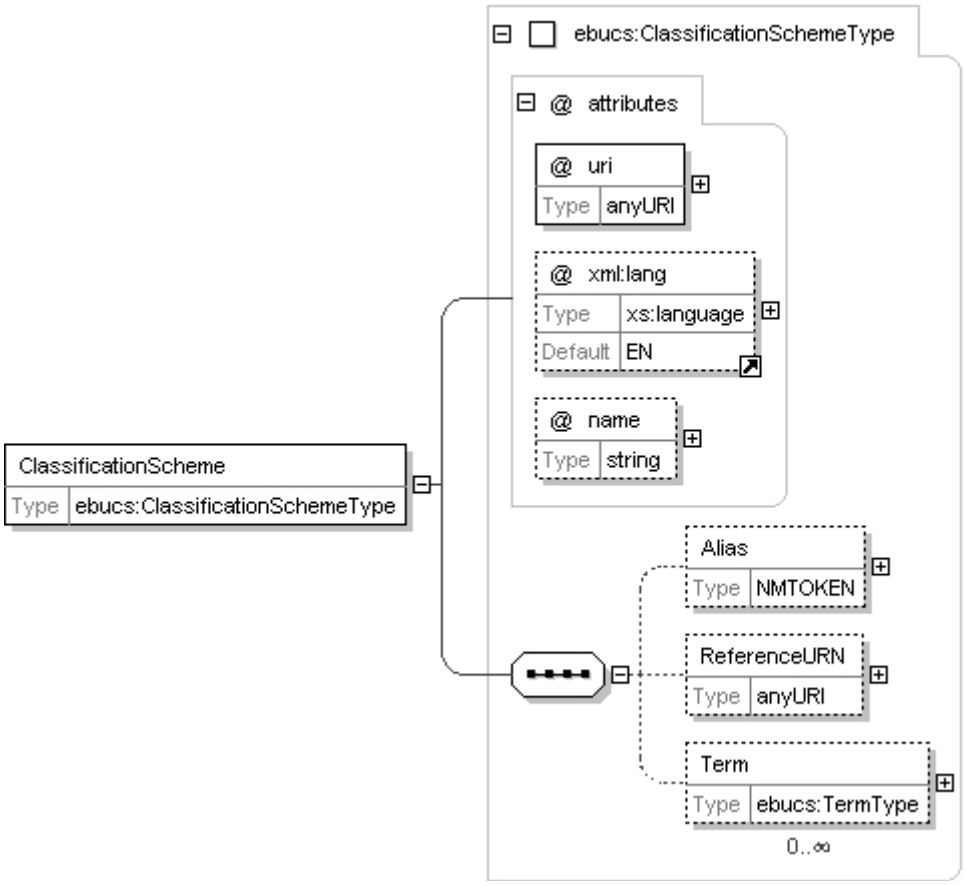
Reference Data and Classification Scheme (CS) namespaces:

- urn:ebu:metadata-cs:ClassificationSchemeName_YYYYMMDD (Month and Day are optional)
  Example: urn:ebu:metadata-cs:RoleCS_20080701

2.3 Classification Scheme

All the terms of a vocabulary are grouped in a structured collection of reference data so-called 'classification scheme'.

```
<element ClassificationScheme type=ebucs:ClassificationSchemeType>
```




```
<complexType name="ClassificationSchemeType">
       <sequence>
              <element name="Alias" type="NMTOKEN" minOccurs="0"/>
              <element name="referenceURN" type="anyURI" minOccurs="0"
              maxOccurs="1"/>
              <element name="Term" type="ebucs:TermType" minOccurs="0"
              maxOccurs="unbounded"/>
       </sequence>
       <attribute name="uri" type="anyURI" use="required"/>
       <attribute ref="xml:lang" default="EN" use="optional"/>
       <attribute name="name" use="optional"/>
</complexType>
```

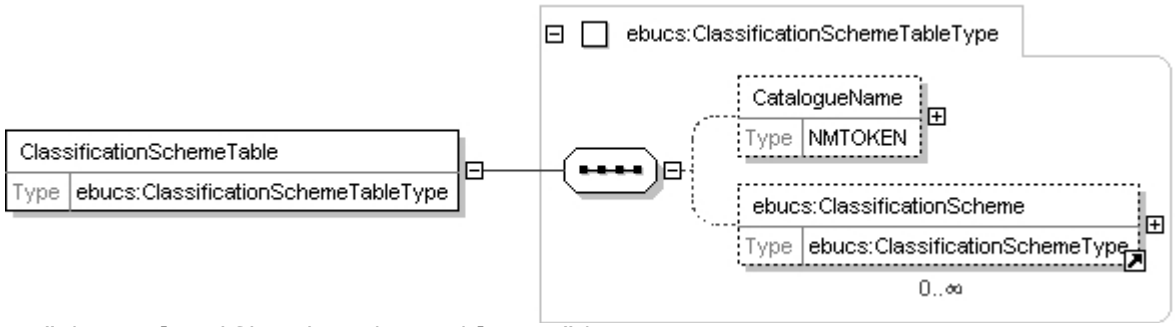
Name	Definition
ClassificationScheme	An element of complex type ClassificationSchemeType used to collect a hierarchical list of terms in a structured reference vocabulary
ClassificationSchemeType	A complex type defining the structure of a hierarchical classification scheme
Alias	An optional element to provide a short name to qualify the classification scheme, which can be used in substitution to the classification scheme URI or URN/URL optionally defined by the ReferenceURN element. Example: ReferenceURN: http://www.ebu.ch/metadata/cs/ebu_ContentGenreCS.xml Alias=" EBUContentGenre" "News / Pure information" is identified by "EBUContentGenre#3.1.1"
ReferenceURN	An element to provide the location of the Classification Scheme for dereferencing. This is typically a URL.
Term	A collection of one or more element constituting the vocabulary defined by the classification scheme
@uri	An attribute to define the Unique Resource Identifier of the classification scheme, which uniquely identifies a version of the classification scheme. This can take the form of a simple name following NCName restrictions.
@xml:lang	An attribute to define the language in which the vocabulary is being expressed (if globally applicable i.e. if only one vocabulary is used)
@name	An optional attribute to define a name for the classification scheme

NOTE: the URI used to identify EBU Classification Schemes are compliant with RFC 5174 – EBU Namespaces.

NOTE: each Classification Scheme defined by a URI should be available as a web resource with a unique resource Locator (URL). More details on the use of the URL are provided in section '3 - Implementation guidelines'.

One or more classification schemes can be grouped to form a catalogue for a particular application, which is defined by its catalogue name (CatalogueName).

```
<element name="ClassificationSchemeTable"
```



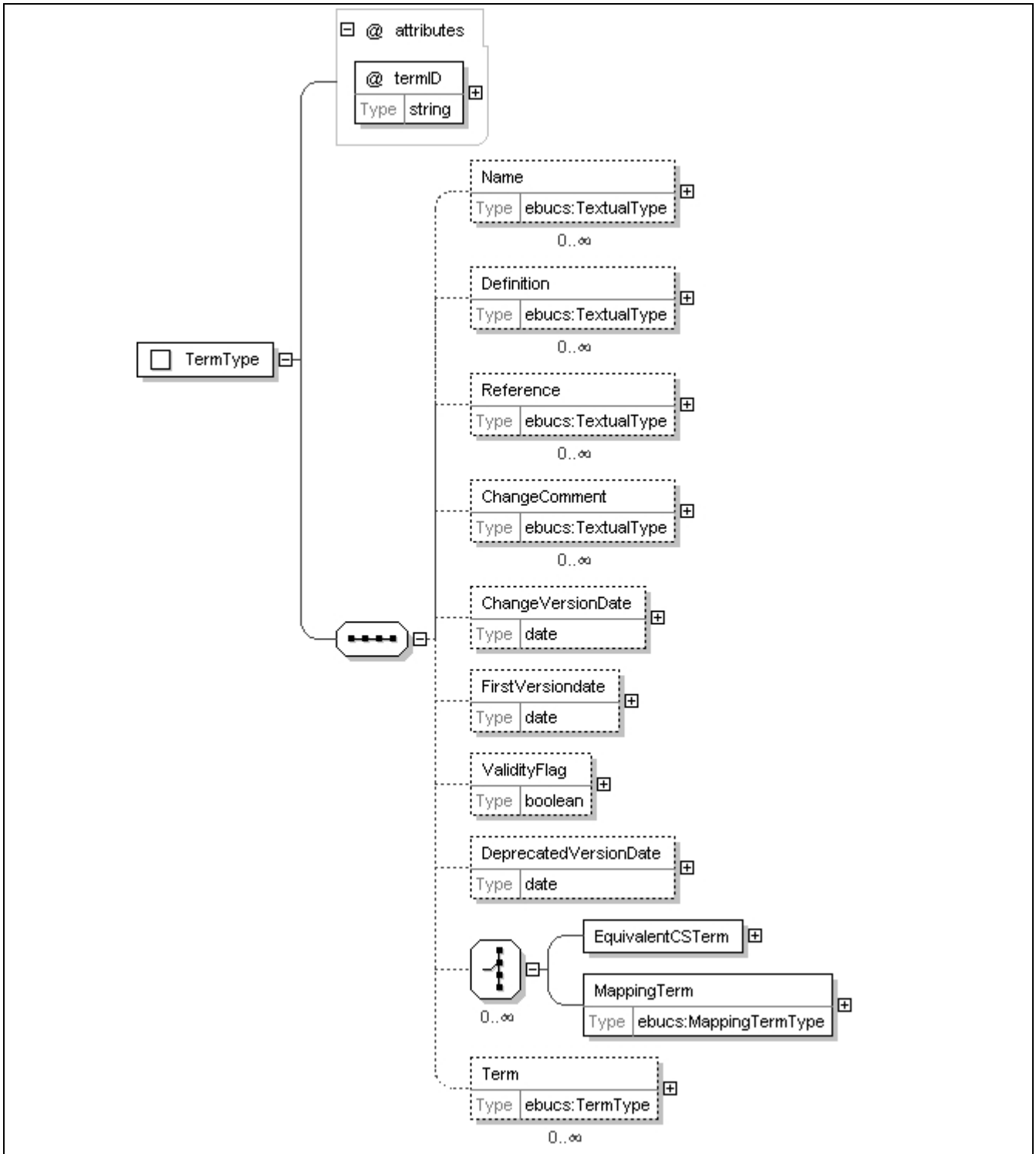

```
 type="ebucs:ClassificationSchemeTableType"/>

      <complexType name="ClassificationSchemeTableType">
            <sequence>
                  <element name="CatalogueName" type="NMTOKEN" minOccurs="0"/>
                  <element ref="ebucs:ClassificationScheme" minOccurs="0"
                  maxOccurs="unbounded"/>
            </sequence>
      </complexType>
```

Name	Definition
ClassificationSchemeTable	An element of complex type ClassificationSchemeTableType used to gather one or more classification schemes in a catalogue
ClassificationSchemeTableType	A complex type to define a table of classification schemes
CatalogueName	An element to give a name to a catalogue
ClassificationScheme	An element to provide one or more occurrences of a classification scheme in the catalogue

2.4 Term

Each term within a hierarchical vocabulary is defined as follows.



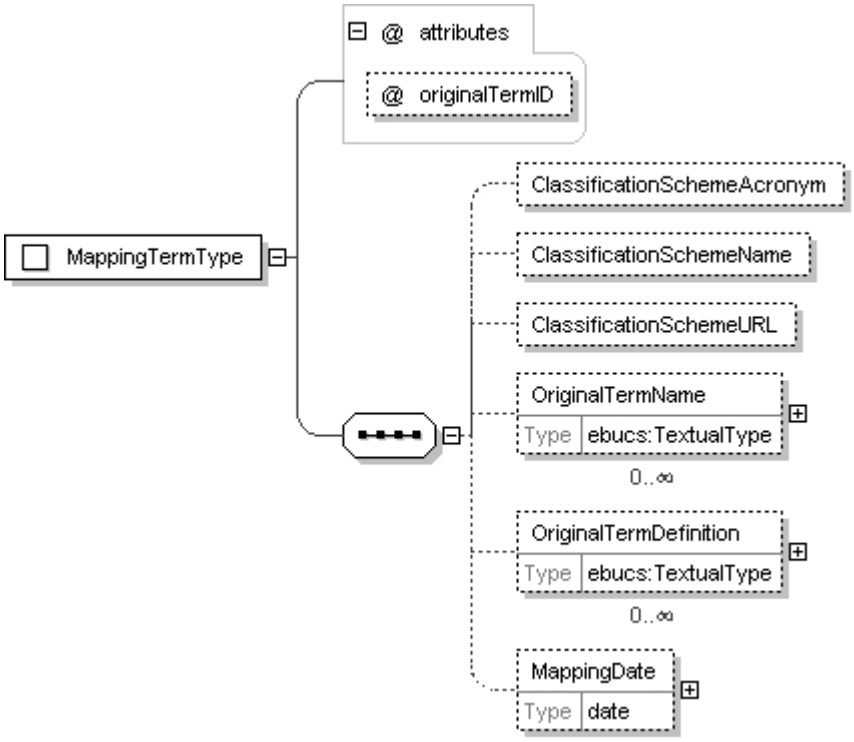

```
<complexType name="TermType">
      <sequence>
            <element name="Name" type="ebucs:TextualType" minOccurs="0"
            maxOccurs="unbounded"/>
            <element name="Definition" type="ebucs:TextualType" minOccurs="0"
            maxOccurs="unbounded"/>
            <element name="Reference" type="ebucs:TextualType" minOccurs="0"
             maxOccurs="unbounded"/>
            <element name="ChangeComment" type="ebucs:TextualType"
```

```
                    minOccurs="0" maxOccurs="unbounded"/>
                    <element name="ChangeVersionDate" type="date" minOccurs="0"/>
                    <element name="FirstVersiondate" type="date" minOccurs="0"/>
                    <element name="ValidityFlag" type="boolean" minOccurs="0"/>
                    <element name="DeprecatedVersionDate" type="date" minOccurs="0"/>
                    <choice minOccurs="0" maxOccurs="unbounded">
                      <element name="EquivalentCSName>
                        <complexType>
                          <attribute name="href" type"anyURI"/>
                        </complexType>
                      </element>                <element name="MappingTerm"
type="ebucs:MappingTermType"/>
                    </choice>
                    <element name="Term" type="ebucs:TermType" minOccurs="0"
                    maxOccurs="unbounded"/>
             </sequence>
             <attribute name="termID" type="string" use="required"/>
      </complexType>
```

Name	Definition
TermType	An complex type to instantiate a term of a vocabulary in a ClassificationScheme
Name	An element of TextualType (see schema) to provide a human readable name for the vocabulary term, which can be expressed in one or more different languages
Definition	An optional element of TextualType (see schema) to provide a human readable definition for the vocabulary term, which can be expressed in one or more different languages
Reference	An optional element to provide a reference to a vocabulary of document from which the term is originating
ChangeComment	An optional element to provide explanations on the reason why the description of a term has been changed
ChangeVersionDate	The date at which the current version of the term description has been published
FirstVersionDate	The date at which an term was first created in the vocabulary
ValidityFlag	A boolean flag to indicate if the term is valid or has been deprecated
DeprecatedVersionDate	The date when the term has been deprecated, if applicable
EquivalentCSTerm	An optional element to identify an equivalent term in another CS with a simple direct URI reference
MappingTerm	An optional element for the instantiation of one or more terms with which the current term can be matched in other vocabularies, accompanied by an exhaustive description of the mapping.
Term	An element of TermType used to instantiate one or more related sub terms in a hierarchical vocabulary structure
termID	An attribute to provide a unique identifier/key for the term within the current vocabulary

2.5 Mapping Term

In addition to a reference to a vocabulary from which a term was originating, it is possible to define a more complete mapping ton terms defined in other vocabularies.




```
<complexType name="MappingTermType">
      <sequence>
            <element name="ClassificationSchemeAcronym"/>
            <element name="ClassificationSchemeName" minOccurs="0"/>
            <element name="ClassificationSchemeURL" minOccurs="0"/>
            <element name="OriginalTermName" type="ebucs:TextualType"
            minOccurs="0"/>
            <element name="OriginalTermDefinition" type="ebucs:TextualType"
            minOccurs="0"/>
            <element name=”MappingDate” type=”date” minOccurs=”0”/>
      </sequence>
      <attribute name="originalTermID" use="optional"/>
</complexType>
```

Name	Definition
MappingTermType	An complex type to instantiate a mapping term element in the description of a vocabulary term of Termtype
ClassificationSchemeAcronym	A short name (alias) for the vocabulary within which the mapping term is defined
ClassificationSchemeName	The full name of the vocabulary within which the mapping term is defined
Classification SchemeURL	The URL at which the mapping term source vocabulary is available (e.g. as schema, webpage or document)
OriginalTermName	The original human readable name by which the mapping term is defined in the source vocabulary
OriginalTermDefinition	The original human readable definition of the mapping term in the source vocabulary
MappingDate	An optional element to provide the date at which the mapping was made
originalTermID	The unique identifier/key by which the mapping term is identified in the source vocabulary

2.6 Example

The following example is extracted from the EBU Content Genre Classification Scheme.

```
 <?xml version="1.0" encoding="UTF-8"?>
<ClassificationScheme uri="urn:ebu:metadata-cs:ContentGenreCS_2009">
   <Alias>EBUContentGenre</Alias>
   <Term termID="3.0">
      <Name xml:lang="en">Proprietary</Name>
      <Definition xml:lang="en">For use where proprietary extensions are required, or the
use of keywords that do not fit in any classification below</Definition>
      <Reference/>
      <ChangeComment>First version after conversion of P/META into XML</ChangeComment>
      <ChangeVersionDate>2007-04-12</ChangeVersionDate>
      <ValidityFlag>1</ValidityFlag>
      <MappingTerm originalTermID="3.0">
         <ClassificationSchemeAcronym>TVA</ClassificationSchemeAcronym>
         <ClassificationSchemeName>TV-Anytime, ETSI TS 102 822, ContentCS
         </ClassificationSchemeName>
         <ClassificationSchemeURL>http://www.ebu.ch/cs/tva/ContentCS.xml#
         </ClassificationSchemeURL>
         <OriginalTermName xml:lang="en">Proprietary</OriginalTermName>
         <OriginalTermDefinition xml:lang="en">For use where proprietary extensions are
         required, or the use of keywords that do not fit in any classification below
         </OriginalTermDefinition>
  <MappingDate>2008-12-03</Mappingdate>
      </MappingTerm>
   </Term>
   …
</ClassficationScheme>
```

3. Implementation Guidelines

3.1 General remarks

Creating, managing, using and maintaining reference data sets and vocabularies requires to follow a minimum of rules, which will be defined in the following sections

For the sake of flexibility, the schema has been designed with all elements and attributes being optional. However, it is expected that identifiers/keys, term names and URI will be provided.

3.2 Good practice for term identifiers / keys

Terms in structured vocabularies such as classification schemes defined in the current specifications are often access through their identifier/key.

There should be only one unique identifier/key per term within a vocabulary.

Hierarchical structures should be logically reflected in the attribution of identifiers/keys, for example:

3.1 NON-FICTION / INFORMATION
3.1.1 News / Pure information
Etc.

Identifiers/ keys allow a language independent common use of vocabularies:

- Term names are provided in different languages under the same term identifier within the same vocabulary
- Equivalent term names are provided in different languages under the same identifier within another related vocabulary identified by a different namespace.

Some vocabularies do not attribute identifiers/ keys to their terms, which is a problem when mapping information is transformed in an alternative format like SKOS. It is strongly advised to attribute such 'dummy' identifiers/keys, when misiing.

3.3 Good practice for term names

It is desirable to allocate simple names to terms, for examples:

```
<Term termID="3.1.1.1.1">
  <Name xml:lang="en">Discussion</Name>
    Etc.
```

However, it is sometimes difficult to achieve, in which case it is recommended to decompose compound names in alternative simple names, for example:

```
<Term termID="3.1.1.7">
  <Name xml:lang="en">Economy/Market/Financial/Business</Name>
  <Name xml:lang="en">Economy</Name>
  <Name xml:lang="en">Market</Name>
  <Name xml:lang="en">Financial</Name>
  <Name xml:lang="en">Business</Name>
```

This is particularly useful to identify mapping with other vocabularies.

3.4 Good maintenance practice

The following rules have been defined for the maintenance of the EBU Classification Schemes.

After a vocabulary has been first published, terms can be added or modified but never removed:

- In the case of the duplication of a term identifier/key, the identifier/key of the duplicates should be corrected without modifying the entire structure of the vocabulary. A comment should clearly remind of the original key, which was mistakenly attributed to a term.
- If a term is obsolete, it should be marked as 'deprecated' with and appropriate date of deprecation. The validity flag should be switched to 'false'.
- If a modification has an impact on part of the vocabulary hierarchical structure, it is advised to deprecate the existing structure and recreate a new arborescence in the vocabulary. Comments should allow users to easily locate where the new structure has been created.

3.5 Good practice in referencing terms

Classification schemes must be available as resources on the open Internet via maintained URLs. In the case of the EBU Classification Schemes, all resources are available under the http://www.ebu.ch/metadata/cs/ root.

Each EBU Classification Scheme is uniquely identified by its URL and namespace (a URI defined in the header of each Classification Scheme in compliance with RFC 5174 – EBU Namespace).

NOTE: It is important to note that if the URL is permanent, the version of the Classification may evolve and is uniquely identified by its URI.

The EBU recommends that the reference to Classification Schemes terms from within content description metadata instances is made in a time persistent manner using the URL of the corresponding Classification Scheme, with the following syntax:

URL '+' # '+' termID

e.g. http://www.ebu.ch/metadata/cs/ebu_ContentGenreCS.xml#3.1

The use of '#' allows a conforming parser to resolve the URL to the appropriate resource (e.g. a webpage, Classification Scheme or document) within which the identifier can be resolved. This method to resolve a term within a resource is left to the appreciation of each recipient. Once the termID has been resolved, the term name can be accessed (e.g. 'News' in the above example).

URLs can be replaced by aliases to provide a more concise, application-specific way of referring to classification terms as long as a look-up table is provided describing the relationship between Aliases and URIs.

Example: If 'GenreCS' replaces http://www.ebu.ch/metadata/cs/ebu_ContentGenreCS.xml", in the above example 'News' will be accessed through "GenreCS#3.1".

4. Maintenance

The EBU Core Metadata Set is maintained by the EBU and suggestions for corrections or additions can be made by mailing to (metadata@ebu.ch). EBU members can also provide feedback via the EBU Technical Department's website:

(http://tech.ebu.ch/MetadataMaintenanceSpecifications).

Contributions will be subject to peer review by the metadata experts participating in P/MAG (http://tech.ebu.ch/groups/pmag), a specialised Project Group of the Production Management Committee (PMC) (http://tech.ebu.ch/groups/details/pmc).

5. Download Zone

Filename	Doc. description	Contents
http://www.ebu.ch/metadata/schemas/EBUCS/2011/EBU_CS_2011.zip	Schema	EBU_CS.xsd, xml.xsd
http://www.ebu.ch/metadata/cs/EBU_cs_p.zip	EBU Classification Schemes	Updated list of EBU Classification Schemes in the ebu_cs format

NOTE: The Simple Dublin Core Schema modified to set the language by default to UK-English.

6. Useful links

EBU Metadata (http://tech.ebu.ch/Metadata)

IETF RFC5174 (EBU namespace): http://tools.ietf.org/html/rfc5174

IANA MIME Type: http://www.iana.org/assignments/media-types/

ISO (http://www.iso.org)

ISO 4217 - Currency codes:
http://www.iso.org/iso/en/prods-services/popstds/currencycodeslist.html

ISO 3166-1 - Country codes (English):
http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html

ISO 3166-1 - Country codes (French):
http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-fr1.html

ISO 639 - Language codes: http://www.loc.gov/standards/iso639-2/

Thesaurus of Geographic Names: http://www.getty.edu/research/tools/vocabulary/tgn/index.html

Sign languages: http://www.signwriting.org/archive/docs1/sw0033-Sign-Language-Codes.pdf

United Nations - Territory codes: http://unstats.un.org/unsd/methods/m49/m49regin.htm

Parental Guidance: http://www.parentalguide.org/, http://www.mpaa.org/
IPTC: http://www.iptc.org
IOC - International Olympic Committee: http://www.olympic.org/uk/sports/
MPEG: http://www.chiariglione.org/mpeg, www.mpeg.org
PBCore: http://www.pbcore.org
TV-Anytime: http://www.etsi.org
DVB: http://www.dvb.org

7. Bibliography

- **EBU Technical Information I36-2003** - Metadata Implementation considerations for Broadcasters
- **EBU Tech 3293** – EBU Core
- **EBU Tech 3295** – P_META Metadata Library
- **EBU Tech 3331** – EXCHANGE
- **EBU Tech 3332** - MUSIC

[*] Page intentionally left blank. This document is paginated for two sided printing

Annex A: EBU Classification Scheme Schema

The schema is available for download from the download zone in § 5, with its accompanying XML stub.

[*] Page intentionally left blank. This document is paginated for two sided printing

Annex B: SKOS Transformation

SKOS (http://www.w3.org/TR/2008/WD-skos-reference-20080829/) , also known as Simple Knowledge Organisation System, has been developed to facilitate the mapping across thesaurus and controlled vocabularies, such as classification schemes, available across the Internet. Its target application is the semantic web.

More information on SKOS is available from the SKOS reference (http://www.w3.org/TR/skos-reference/) and the SKOS Primer (http://www.w3.org/TR/skos-primer/).

The following XSLT transformation sheet defines all the EBU classification schemes' terms and the terms mapped from external vocabularies as concepts each identified by a unique URI (URL#termID). It also defines the relationships needed to reconstruct the hierarchy of EBU Classification Schemes. Finally, it establishes relationship to mapping terms (generalised as narrow matches).

The 'Result_File_Name' (highlighted in yellow in the schema) should be changed to adapt to the Classification Scheme being transformed. For example, the SKOS transform of ebu_ContentGenreCS.xml will be called ebu_ContentGenreCS.skos.xml

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

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#">
    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>

    <xsl:template name="TopConcept">
        <xsl:param name="AliasName"/>
        <xsl:variable name="TopConceptID" select="concat($AliasName,'#',@termID)"/>
        <skos:HasTopConcept>
            <rdf:Description rdf:about="{$TopConceptID}">
                <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
            </rdf:Description>
        </skos:HasTopConcept>
    </xsl:template>

    <xsl:template name="Concept">
        <xsl:param name="AliasName"/>
        <xsl:variable name="conceptID" select="concat($AliasName,'#',@termID)"/>
        <rdf:Description rdf:about="{$conceptID}">
            <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
            <skos:prefLabel xml:lang="en">
                <xsl:value-of select="Name"/>
            </skos:prefLabel>
            <skos:definition xml:lang="en">
                <xsl:value-of select="Definition"/>
            </skos:definition>
            <!--skos:inScheme rdf:resource="{$AliasName}"/-->
            <xsl:variable name="BroaderConceptID"
                select="concat($AliasName,'#',parent::Term/@termID)"/>
            <xsl:if test="string-length(substring-after($BroaderConceptID,'#'))!=0">
                <skos:broader>
                    <rdf:Description rdf:about="{$BroaderConceptID}">
                        <rdf:type
              rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
                    </rdf:Description>
                </skos:broader>
            </xsl:if>
            <xsl:for-each select="child::Term">
```

```
            <skos:narrower>
                <xsl:variable name="narrowerConceptID"
       select="concat($AliasName,'#',@termID)"/>
                <rdf:Description rdf:about="{$narrowerConceptID}">
                    <rdf:type
 rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
                </rdf:Description>
            </skos:narrower>
        </xsl:for-each>
        <xsl:for-each select="Name[position()>1]">
            <skos:hiddenLabel>
                <xsl:variable name="other_name"
                    select="translate(current(),'ABCDEFGHIJKLMNOPQRSTUVWXYZ/-
()','abcdefghijklmnopqrstuvwxyz/-()')"/>
                <xsl:value-of select="$other_name"/>
            </skos:hiddenLabel>
        </xsl:for-each>
        <skos:changeNote>
            <xsl:value-of select="ChangeComment"/>
        </skos:changeNote>
        <skos:historyNote>
            <xsl:value-of select="ChangeVersionDate"/>
        </skos:historyNote>
        <skos:example/>
        <xsl:if test="ValidityFlag=1">
            <skos:note>Valid</skos:note>
        </xsl:if>
        <xsl:if test="ValidityFlag!=1">
            <skos:note>Obsolete</skos:note>
        </xsl:if>
        <xsl:for-each select="MappingTerm">
            <xsl:variable name="matchID"
                select="concat(ClassificationSchemeURL,@originalTermID)"/>
            <skos:narrowMatch>
                <rdf:Description rdf:about="{$matchID}">
                    <rdf:type
 rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
                </rdf:Description>
            </skos:narrowMatch>
        </xsl:for-each>
    </rdf:Description>
</xsl:template>

<xsl:template name="mappingConcept">
    <xsl:variable name="mappingConceptID"
        select="concat(ClassificationSchemeURL,@originalTermID)"/>
    <rdf:Description rdf:about="{$mappingConceptID}">
        <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
        <skos:prefLabel xml:lang="en">
            <xsl:value-of select="OriginalTermName"/>
        </skos:prefLabel>
        <skos:definition xml:lang="en">
            <xsl:value-of select="OriginalTermDefinition"/>
        </skos:definition>
        <skos:inScheme rdf:resource="{ClassificationSchemeURL}"/>
    </rdf:Description>
</xsl:template>

<xsl:template match="/">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:skos="http://www.w3.org/2004/02/skos/core#">
        <xsl:variable name="alias"
        select="'http://www.ebu.ch/metadata/cs/skos/Result_File_Name.skos.xml'"/>
        <rdf:Description rdf:about="{$alias}">
            <rdf:type
rdf:resource="http://www.w3.org/2004/02/skos/core#ConceptScheme"/>
            <xsl:for-each select="ClassificationScheme/Term">
                <xsl:call-template name="TopConcept">
                    <xsl:with-param name="AliasName" select="$alias"/>
                </xsl:call-template>
            </xsl:for-each>
```

```
            </rdf:Description>
            <xsl:for-each select="descendant-or-self::Term">
                <xsl:call-template name="Concept">
                    <xsl:with-param name="AliasName" select="$alias"/>
                </xsl:call-template>
            </xsl:for-each>
            <xsl:for-each select="descendant-or-self::Term/MappingTerm">
                <xsl:call-template name="mappingConcept"/>
            </xsl:for-each>
        </rdf:RDF>
    </xsl:template>

</xsl:stylesheet>
```