Controlled vocabularies in XML documents

Notes and recommendations by Stephen M. Richard
US Geoscience Information Network
Wednesday, January 20, 2010

Introduction

The use of controlled vocabularies (codelist) in xml documents is essential for semantic interoperability. Controlled vocabulary terms are used to specify terminological attribute values in xml instance documents. These attributes may have a variety of relationships to the data:

1. they may specify a data type for soft-typing an element, with implications for cardinality, existence or value domain of other attributes (e.g. GeologicUnitType in GeoSciML)
2. act as quantifiers for properties of a described real-world entity (e.g. what color is the car I'm selling)
3. Categorize a resource for indexing purposes, as in keywords.

Following the conventions from W3C SKOS (http://www.w3.org/2004/02/skos/; see also https://www.seegrid.csiro.au/subversion/CGI_CDTGVocabulary/trunk/Documents/Use_SKOS_encodingVocabularies.pdf :
The controlled vocabulary is a collection of concepts, each with an identifier and one or more labels. Each label is a word in some language that conveys the concept to a speaker of that language. Because of the fuzziness of human language, it is usually necessary to include a definition that explicitly scopes the concept; this may use common language (and may be represented in one or more languages), or may be expressed using formal logic of some sort (e.g. OWL).

Here is a list of the properties that are useful for information interchange
1. an identifier for a concept. Ideally this is globally unique; at a minimum it must be unique within the scope of its containing vocabulary
2. an identifier for the controlled vocabulary within which the concept is defined
3. one or more language-localized labels for the concept for user presentation.
4. the location of one or more services that can operate on identifiers or labels used in 1, 2, and 3 to provide added value, e.g. a language-localized label or definition for the concept, a definition using some formal ontology scheme (e.g. OWL), lists of related concepts (e.g. broader, narrower, related, equivalent in another vocabulary, subsuming, subsumed...).

Machine information processing is done operationally with the identifiers (using 1 or 1 and 2), or by accessing formal definitions by resolving the identifier against some registry (using 4). Human interpretation of the content is based on the concept labels (using 3), perhaps accessing text definitions (using 4).

While the basic properties required for inclusion of a controlled vocabulary term in a document are widely recognized and included in xml schema in a variety of way, the usage of these representation components varies widely, making interoperability more difficult. Interoperability problems arise when it is not clear as to:

1. what is the identity property for a controlled vocabulary term, and where that identifier is placed in xml instances documents (is it codeListValue?, is it the element value?).
2. if associated codespace or other scoping property is an identifier for a vocabulary, or a URL for a service that can resolve a term identifier; if it is the URL for a service, what will that service provide.
3. What labels are available for the controlled vocabulary term and the language of the label.

Schema elements that currently exist:

Codelist

as currently used in gco:CodeTypes, the codelist mostly seems to be used as a URL for a resource that provides a definition of a codeListValue, consistent with the guidance in ISO19139 (p.29): "codeList attribute contains a URL that references a codeList definition within a registry or a codelist catalogue". Unfortunately, this does not allow identification of the codelist that is in use—different provides reference different URL's for looking up definitions of codelistValues. If different codeList vocabularies are in use (e.g. ISO and NAP) there is no reliable way to construct an xpath to determine which vocabulary is in use.

Codespace

guidance in ISO19139 is "codeSpace attribute is an optional identifier (URI); when present it refers to the alternative expression of the codelist value definition effectively expressed as the value of the element", whereas in GML 3.2.1 "the value of the codeSpace attribute shall indicate a dictionary , thesaurus, classification scheme, authority, or pattern for the term."

xlink:simpleLink

The xlink:simpleLink attribute group includes these attributes:
xlink:href -- URI reference as defined in [IETF RFC 2396]; supplies the data that allows an XLink application to find a remote resource (or resource fragment)
xlink:role -- URI reference as defined in [IETF RFC 2396]; identifies some resource that describes the intended property
xlink:arcrole -- URI reference as defined in [IETF RFC 2396]; identifies some resource that describes the intended property
xlink:title -- used to describe the meaning of a link or resource in a human-readable fashion

xlink:show-- used to communicate the desired presentation of the ending resource on traversal from the starting resource. Value must be one of the values "new", "replace", "embed", "other", and "none".
xlink:actuate -- one of the values "onLoad", "onRequest", "other", and "none".

The xlink:simpleLink attribute group is implemented in GML through AssociationAttributeGroup XML attribute group, and in ISO19139 through the ObjectReference XML attribute group

CI_OnlineResource

ISO19139 defines gmd:CI_OnlineResource, which provides an UUID attribute, and child elements: URL (required), 'function' from a controlled vocabulary, and free-text protocol, applicationProfile, name, and description elements. URL, function, and name appear to map to equivalent attributes href, role, and title in xLink. The semantics of protocol and applicationProfile in CI_OnlineResource are not clearly defined and these elements are used differently by different metadata practitioners.

gmd:descriptiveKeywords/MD_Keywords

descriptiveKeywords element has xlink:SimpleLink attribute group.
MD_Keywords content of descriptiveKeywords provides for 1..* keyword character strings, one type from a keyword type code list, and a thesaurusName element (with xlink:SimpleLink attribute group) with CI_Citation content. CI_OnlineResource is a child of CI_Citation through citedResponsibleParty, thus not really applicable to linking to semantic resources associated with the keywords.
The keyword element is typed CharacterString, with the implication that the keywords are language-localized words, not identifiers for vocabulary concepts. The ISO keyword type code list is {discipline, place, stratum, temporal, theme}.

Proposed solution

Codelists

The gco:CodeListValue_Type contains the basic elements required, except for language localization. The names of the attributes could be changed to make their semantics clearer
codeList == resourceLookup ==xlink:href: a URL that provides some human-applicable resource that defines the term—html page, image, etc… Of course, this duplicates the intent of xlink:href, so adding the xlink:simpleLink attribute group would achieve this objective without creating a new attribute (and confusion about what it means)… and provide the added possibility of using some other xlink attributes like role and title.
codeListValue == termID: a language neutral identifier, preferably conforming to some URI scheme (not a plain English word, at least put in a URN form e.g. urn:ISO:19139:code:created)
codeSpace == vocabularyID: the name of this attribute could stay the same, just need to be clear that usage follows gml provision—it is an identifier for a thesaurus, vocabulary,

authority that is the scope for the termID. If multiple vocabularies are in use, this is the property used to determine which vocabulary a term instance belongs to.

Just have to add **locale** as in gmd:LocalisedCharacterString—a URI that identifies the language of the codeList element content. Following the language localization pattern already in place, this property could be optional, added only if the metadata is multilingual, otherwise the element value strings would be assumed to be in the language specified by MD_Metadata/language.

So the proposal would be to add a new type in the CharacterString substitution group:

```
<complexType name="ScopedLocalizedTerm_Type">
        <simpleContent>
                <extension base="xs:string">
                        <attributeGroup ref="xlink:simpleLink"/>
                        <attribute name="vocabularyID" type="xs:anyURI"
                            minOccurs="0"/>
                        <attribute name="termID" type="xs:anyURI" minOccurs="0"/>
                        <attribute name="locale" type="xs:anyURI" minOccurs="0"/>
                </extension>
        </simpleContent>
</complexType>

<xs:element name="ScopedLocalizedTerm"
            type=" ScopedLocalizedTerm_Type "
            substitutionGroup= "gco:CharacterString"/>

<xs:complexType name=" ScopedLocalizedTerm _PropertyType">
                <xs:sequence minOccurs="0">
                        <xs:element ref="ScopedLocalizedTerm "/>
                </xs:sequence>
                <xs:attribute ref="gco:nilReason"/> <!--maybe this shouldn't be here—
            does it make sense to include an empty term filler? -->
            </xs:complexType>
```

Examples:

```
<ScopedLocalizedTerm  xlink:href="http://url.for.term.definition/english.html
            vocabularyID="urn.USGIN.metadata.keywords.theme"
            termID="urn.USGIN.terms.3463278"
            locale="eng">igneous rock</ScopedLocalizedTerm>

<ScopedLocalizedTerm  xlink:href="http://url.for.term.definition/espanol.html"
            vocabularyID="urn.USGIN.metadata.keywords.theme"
            termID="urn.USGIN.terms.3463278"
            locale="spa"> roca ígnea</ScopedLocalizedTerm>
```

<ScopedLocalizedTerm xlink:href=”http://url.for.ontologyService/owl?termid=343278”
xlink:role=”OWL_definition”
vocabularyID=”urn.USGIN.metadata.keywords.theme”
termID=”urn.USGIN.terms.3463278”
locale=”rus”> магматическая порода</ScopedLocalizedTerm>

Hmmm, do we need to worry about character encoding?

Keywords

The request is to implement a scheme for categorization of keyword terms to particular classes, applicable for resource discovery.

1. **Add MD_Keywords/class property, 0..***
Generalization of the Keyword type concept, with the type being populated by a generalized scoped concept (ScopedLocalizedTerm). Each MD_Keywords instance would be restricted to keywords that all have the same class categorization. Could change defiition of type element, but adding new element seems less likely to break things.

2. **Add 0..* CI_OnlineResource elements in MD_Keywords**.
In order to account for binding between a concept instance and services that provide resources related to the concept (definition, translation, related concepts). Online resource could be link to metadata record for service, link to provide definition, or other specific operations using the keyword identifier

3. **Keywords should be ScopedLocalizedTerm (see codelist, above)**. Since ScopedLocalizedTerm as defined above is in CharacterString substitution group, this could be implemented using xsi:type=”ScopedLocalizedTerm_Type” in a CharacterString element, following pattern for extension of character string in ISO19139.

Leave thesaurusName and type elements unchanged for backward compatibility. The identifier for a thesaurus (CI_Citation/identifier/MD_Identifier) should contain the same identifier as the vocabularyID on the ScopedLocalizedTerms; inclusion of a reference to the CI_Citation would provide more complete information on the vocabulary, thesaurus, authority etc. that defines/maintains the vocabulary.

Schema:
<xs:complexType name="MD_Keywords_Type">
<xs:complexContent>
<xs:extension base="gco:AbstractObject_Type">
<xs:sequence>
<xs:element name="keyword" type="ScopedLocalizedTerm_PropertyType"
maxOccurs="unbounded"/>
<xs:element name="type" type="gmd:MD_KeywordTypeCode_PropertyType"
minOccurs="0"/>

```
        <xs:element name="thesaurusName" type="gmd:CI_Citation_PropertyType"
                minOccurs="0"/>
        <xs:element name="class" type="ScopedLocalizedTerm_PropertyType"
                minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="linkedResource" type="gmd:CI_OnlineResource"
                minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                </xs:extension>
          </xs:complexContent>
</xs:complexType>
```

Background notes

GML defines a basic type gml:CodeType and gml:CodeWithAuthorityType in schemas.opengis.net/gml/3.2.1/basicTypes.xsd. The code type "adds an XML attribute codespace to a term, where the value of the codeSpace attribute (if present) shall indicate a dictionary, thesaurus, classification scheme, authority, or pattern for the term. The codespace attribute is required in CodeWithAuthorityType, optional in CodeType.

GML 3.2.1 defines an AssociationAttributeGroup that consists of the gml:nilReason attribute and the xlink:simpleLink attribute group (in gmlBase.xsd)

ISO19139 (gcoBase.xsd) defines similar attribute group named 'ObjectReference' in the gco namespace:

```
<xs:attributeGroup name="ObjectReference">
        <xs:attributeGroup ref="xlink:simpleLink"/>
        <xs:attribute name="uuidref" type="xs:string"/>
</xs:attributeGroup>
```

This attribute group is assigned to elements in the metadata schema to enable inclusion by reference.

xlink:show-- used to communicate the desired presentation of the ending resource on traversal from the starting resource. Value must be one of the values "new", "replace", "embed", "other", and "none".

- New—load resource in new window.
- Replace—replace current window content.
- Embed--load its presentation in place of the presentation of the starting resource; presentation of the starting resource typically does not consist of an entire document; it would be the entire document only when the root element of the document is a simple link. Thus, embedding typically has an effect distinct from replacing.
- Other--behavior of an application traversing to the ending resource is unconstrained by this specification; other markup present in the link should determine the appropriate behavior

- None--application traversing to the ending resource is unconstrained by this specification; No other markup is present to help the application determine the appropriate behavior