Non-information resources in the world wide web

Web resources describe entities in the real world

Controlled vocabularies in XML documents

Stephen M. Richard, revised based on e-mail discussion with Simon Cox, Roberto Lucchi, Clemens Portele, Marten Hogeweg, Guillaume Declaux, and Agnès Tellez-Arenas

2010-02-18 expand to general discussion of reference to non-information resources on the web.

Tuesday, February 09, 2010

Introduction

The use of controlled vocabularies in xml documents is essential for semantic interoperability. As used here, a controlled vocabulary is defined as a collection of concepts. Controlled vocabulary terms representing individual concepts are used to specify terminological values in xml instance documents. These values 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 entity (e.g. what color is the car I'm selling, what is the format of a resource)
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 some 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 such that a compound key can be constructed with a vocabulary identifier and the concept identifier.
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. locators (URL's) for 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 machine-interpretable definitions by dereferencing 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 concept in a document are widely recognized and included in xml schema in a variety of ways, 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 concept, and where that identifier is placed in xml instances documents? (is it an element attribute like codeListValue?, is it the element value?).
2. Is an associated codespace or other scoping property an identifier for a vocabulary, or a URL that dereferences the identifier? If it is the URL, what kind of document will be returned by ‘getting’ that URL?
3. What labels are available for the controlled vocabulary concept? Are the labels language localized?

Dereferencing, information resource, and non-information resources, representation

A dereferencable URI is one that can function as a locator. In other words it is a URI using a protocol that specifies how to fetch a document based on the URI. HTTP is the only(?) widely used protocol in this capacity, because a standardized dereferencing process (the Domain Name System) is implemented as part of the World Wide Web infrastructure. The urn: protocol does not specify how to find documents with URN URIs, so they are not dereferencable without some external implementation of a dereferencing scheme. Standard, off the shelf web browser software will not access a document from the web given a URI using a non-dereferencable protocol.

An “information resource” is something that can be transmitted electronically. Documents, such as web pages, RDF/XML documents, and binary files are all information resources. “Non-information resources” are those resources that can be identified by a URI but which cannot be transmitted electronically. Human beings, abstract concepts, etc. are non-information resources. Both information and non-information resources can be identified with URIs. A computer can only display information resources, but they are (by definition) incapable of displaying a non-information resource itself. Non-information resources can only be presented using a ‘representation’, which is an information resource meant to communicate the nature of the non-information resource that corresponds to the URI. Examples of representations include pictures of people, a free text description, or a description using formal syntax like xml or OWL.

According to web architecture standards, the HTTP 200 OK response to get requests is to be used for URLs denoting information resources only. So when an HTTP get is made using a URL URI for a non-information resource, the theory is that the web server at the other end first sends back a HTTP 303 “See Other” response, i.e. a redirect. This indicates that the URL URI represents something that the web server can not provide directly. It sends back instead a URL directing the user agent to an information resource, with the implication that information about the thing identified by the original URL URI can be found in the document at that URL. This behavior does not seem to actually occur very often in practice…

For a federated information system to provide interoperability there must be some understanding of what kind of information resource a redirected URL for a non-information resource will return. HTTP provides mechanisms for content negotiation, so within the framework of resources that can be identified using MIME types, there may be several options for the kind of resource returned, for both information and non-information resources.

Hereafter we'll use the phrase *[URI] in the scheme* to refer to the set of URIs identifiable as part of a naming scheme, and the phrase *access a URI* as shorthand for the retrieval of a representation of the resource identified by a URI.

Issues

Requirements for identifiers

identifiable

The desire for branding to be evident in URIs is both widespread and understandable. Identifiability of a URI means that the agent responsible for the identified resource is communicated to a person seeing the URI. URI identifiability is a form of advertising, where the admittedly modest impact of a single use of an identifiable URI is potentially magnified greatly by widespread replication. Identifiability also is a cornerstone of trust: brand recognition and successful URI access are mutually reinforcing.

useable

It should require little or no effort on the part of ordinary users to retrieve a useful representation of the resource identified by a URI in the scheme.

reliable

It should always be possible to get a positive response (either a representation or other definite advice about the resource) from an attempt to dereference a URI in the scheme.

transparent

Dirk and Nadia have read the TAG finding on opacity of URIs (Mendelsohn and Williams 2007), and are agreed that their scheme should be explicitly transparent, that is, it should be evident what each of their URIs is about just by looking at it. They understand and accept that this means they will have to document the nature of the mapping from URIs to resources as part of their public site description.

distributed

Dirk says "We have to be able to give our members control of the URIs that concern them". Nadia says "Right, for instance, each studio has to be responsible for their own pictures". That is, the owner of a set of URIs in the scheme must be able to **delegate** support for the transfer of naming authority (control over the meaning of URIs) for designated parts of the scheme.

stable

Nadia says "People have to be able to rely on our URIs". Dirk adds "Yes, the meaning of our URIs shouldn't change, no matter what happens". After further discussion, they distinguish three kinds of stability:

- **owner stability** "No-one can take our URIs away from us." That is, ownership of a URI, and the authority over a URI's meaning which follows from it, continues as long as the owner wants it to;

- **resource stability** "For at least some of our URIs, we won't ever change what they are for". That is, in AWWW terms, what *resource* such a URI *identifies* shouldn't change;
- **representation stability** "For at least some of our URIs, we want to guarantee that *exactly* the same page will always come up". That is, the *representation* retrieved from such URIs shouldn't change.

self-describing

"How do we find out the status of one of our URIs", asks Dirk, and goes on "Who its naming authority is, who wrote its representation(s), whether it's meant to be stable or not—oh, lots of stuff"? "There should be a standard place to put metadata, and a standard way to get at it, for every one of our URIs", replies Nadia. That is, given a URI in the scheme it should be possible to retrieve metadata about the URI and the resource it identifies independently of the representation of that resource, if any.

safe

Nadia remembers security: "If we turn out to want to hold or exchange information which is private to some of our members or users, they have to know it's safe." "Right," says Dirk, "We should be able to reliably identify our users and members, and keep our interchanges with identified parties private, without otherwise changing things."

What representations are available

What does dereferencing return

How to implement dereferencing

Dereferencing service

Encode in URI (url)

Schema elements that currently exist:

Review of some current implementation for binding identifiers with information resources.

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 servers provide different URL’s for looking up definitions of codelistValues from the same codelist. The codelist values do not use any kind of globally unique identifier scheme, there is no automatic test to determine if two codelistValues that have different associated codelist URL’s in fact represent the same concept. Also, 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 for the ‘codespace’ attribute 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.” The implication of the GML 3.2.1 definition is that the codespace/codeListValue pair should constitute a globally unique identifier, but this is not explicitly stated.

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.

XLink provides an obvious means to associate a reference using an identifier with some location providing a document describing or defining a non-information resource, or returning an information resource, depending on what the identifier identifies. Within the in scope of xlink, there does not seem to be any way to engineer content negotiation to request different representations available through the href. Role or arcRole could be used give client software an indication of the kind of document located by the href, and title could be used for labeling the identified resource in the computer interface.

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}.

RDFa

Adds attributes to xml elements for RDF encoded metadata, designed for use in XHTML web documents. These allow association of xml element values with URI's for properties, datatypes, linked resources (only one, because they are xml attibutes), types, or identifiers (rdf:about). Similar to xlink, but more possibilities.

Content Negotiation

From RFC2616:

> "Naturally, it is desirable to supply the user with the "best available" entity corresponding to the request. Unfortunately for servers and caches, not all users have the same preferences for what is "best," and not all user agents are equally capable of rendering all entity types. For that reason, HTTP has provisions for several mechanisms for "content negotiation" -- the process of selecting the best representation for a given response when there are multiple representations available."

The HTTP protocol allows the requesting HTTP client to specify the MIME type of what it wants in the HTTP Accept header. The usual case is HTML (text/html) page, but a wide variety of file formats are possible. Content negotiation is a process in which the client and server software agree on the format of the response document.

A web-browsing client accessing an RDF URL URI might want an HTML representation of a resource to display to the user (some textual description of what the URI represents), whereas an RDF client may want an RDF/XML description of the resource. In this case, the client specifies Accept: application/rdf+xml. We think of the HTML page and the RDF/XML document as two information-resource representations of the same non-information resource (i.e. what the URI denotes). HTTP GET allows RDF clients to find triples about a particular resource that has a URL URI. Content negotiation allows there to be both an HTML page and an RDF document at the same URL.

What is not yet clear is how a user constructing a web page can construct a link from a web page to access a particular representation using the http: accept header field with a mime type.

At this point is seems that having URI's for different representations makes the most sense.

Link tags

Another way to associate a URI with an HTML page to an RDF document is to use the HTML link tag in the HTML head section. This tag will help RDF clients discover related RDF information from a page the user may already be browsing. The 'alternate' HTML link type defined in section 6.12 of the HTML 4.01 is the pertinent one in this context. For example, a URL URI might point to an HTML document, with the following link element in its header: `<link rel="alternate" type="application/rdf+xml" href="SkosVocab.rdf"/>`. The file SkosVocab.rdf is expected to be an RDF/XML document, which would be expected to describe the identified non-information resource. To be interoperable, there would have to be some clear understanding by the client to look for such link elements in the document located by the URL URI, and of the format of the linked description. The link @title attribute might be used to provide some indication of this.

Proposed solution

Identifiers need to be easily dereferenceable. Identifiers for information resource should dereference to provide the resource (subject to access constraints if they exist), identifiers for non-information resources should dereference to some canonical form, which would have to be agreed upon and well known within a community of practice to be interoperable.

There might be a variety of representations associated with a resource, e.g. different file formats, or various kind of formal or informal descriptions for non-information resources. MIME types can be used within the HTTP content negotiation framework to distinguish some of these representations, but do not provide sufficient granularity in all cases (SKOS vs OWL, lay-person definition vs. expert definition). A richer typing of resource representations would be useful.

Concept identifiers are defined in the framework of a vocabulary that has some scope and attribute space partitioned by the concepts in the vocabulary. It is important to identify the containing vocabulary within which a concept is defined.

My experience with codelists in ISO19139 and scopeName or ControlledConcept in GeoSciML/GML indicates that more clarity of definition and consistency of usage is necessary to distinguish concept identifiers, labels, vocabulary identifiers, and URL's to dereference an identifier. This might be achieved with some more explicit naming of XML elements, e.g. something like termID, VocabularyID, xlink:href, localizedTermLabel.

Possibilities:

In no particular order, with varying dependencies between the options chosen.

- Use xlink for dereferencing identifiers. Upside--it's an existing scheme in common use. Downside--only one attribute value allowed on an element. Ability to specify type of resource accessed by dereferencing is limited, so there must be agreement on what these canonical forms are, and consistent usage.
- Require identifiers to be URL's that will dereference using standard WWW DNS. Requires strict adherence to use of identical strings for URL such that equivalence can be tested by string comparison, and strong conventions about the canonical form that will be returned on different MIME types. Interesting questions for access constraints given identifiers for protected resources.
- Standardize dereferencing by providing resolver URL to which identifier is appended to create URL that dereferences identifier. Upside-decouple identifier from resolver service, allows local resolution, might provide different resolvers for different kinds of representations; resolvers can relocate without having to deal with redirecting from old address. Can be used with non-http URL identifier schemes (e.g. UUID). Downside--identifier string by itself does not have information to guide dereferencing.
- Concept identifiers might be constructed hierarchically, with a vocabulary identifier prefix and a concept identifier suffix.
- Current CGI (GeoSciML) approach -- vocabularies have one URN, with prefix indicating URN is a classifier scheme (vocabulary); concepts have different URN with prefix indicating identifier resource is a classifier (concept). Although not inherent in this approach, in the CGI scheme the URN for the vocabulary can be calculated from the URN for the concept.

Possible schema for a light weigh xml element representing a concept from a controlled vocabulary:

Schema:

```
<complexType name="ScopedLocalizedTerm_Type">
  <simpleContent>
    <extension base="xs:string">
      <attributeGroup ref="xlink:simpleLink"/>
        <!-- skip vocabularyID if xlink:href URL resource for term has well know
        mechanism to identify containing vocabulary, and this level of indirec-
        tion is acceptable -->
      <attribute name="vocabularyID" type="xs:anyURI" minOccurs="0"/>
      <attribute ref=”rdf:about” />
      <attribute ref=”xml:lang” minOccurs="0"/>
    </extension>
  </simpleContent>
</complexType>
```

Examples:

No xlink:role specified, so convention is that standard canonical form (whatever that is in the application profile) will be returned by href. VocabularyID would come from context (as in Keyword example below), or through convention to include it in the canonical resource document returned by xlink:href. Language is assumed to be language declared for containing document.

```
<ScopedLocalizedTerm
        xlink:href=”http://url.for.ML_term.def?ID=’urn.USGIN.terms.3463278’”
        rdf:about=”urn.USGIN.terms.3463278”
        >igneous rock</ScopedLocalizedTerm>
```

If the xlink:href is not included, assumption is that termID is a URL URI, which will dereference to a standard canonical form document. Similar assumption about vocabulary identifier and language as in last example.

```
<ScopedLocalizedTerm
        rdf:about=”http://resource.usgin.org/urn.USGIN.terms.3463278”
        >igneous rock</ScopedLocalizedTerm>
```

Explicit language localization, and vocabulary identification. Dereferencing of vocabulary identifier is not solved.

```
<ScopedLocalizedTerm
      xlink:href=”http://url.for.term.def?lan=’spa’&ID=’urn.USGIN.terms.3463278’”
      vocabularyID=”urn&#58;USGIN.metadata.keywords.theme”
      rdf:about=”urn.USGIN.terms.3463278”
      xml:lang=”spa”>roca ígnea</ScopedLocalizedTerm>
```

Use xlink:role to specify the kind of resource representation xlink:href will provide.

```
<ScopedLocalizedTerm
        xlink:href=”http://url.for.ontologyService/owl/urn.USGIN.terms.3463278”
        xlink:role=”OWL_definition”
        vocabularyID=”urn&#58;USGIN.metadata.keywords.theme”
        rdf:about=”urn.USGIN.terms.3463278”
        xml:lang=”rus”>магматическая порода</ScopedLocalizedTerm>
```

Comparison to ISO19139 Codelist

The gco:CodeListValue_Type contains the basic elements required for reference concepts from a vocabulary, 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.

Keywords in ISO19139

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

1. **Keyword categories or types**

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.

2. **Add 0..* linkedResource 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). Linked 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**. Since ScopedLocalizedTerm could be defined in gco:CharacterString substitution group, this could be implemented using xsi:type=”ScopedLocalizedTerm_Type” in a gco:CharacterString element, following pattern for extension of character string in ISO19139. This would not break applications that expect a simple CharacterString element.

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. Thus if thesaurus (vocabulary) citation is included, vocabularyID in ScopedNames is not required. Inclusion of a reference to the CI_Citation would provide more complete information on the vocabulary, thesaurus, authority etc. that defines/maintains the vocabulary.

0 to many gmd:CI_OnlineResource elements allow binding to other representations or related resources for keyword concepts (e.g. ontology, usage guidelines).

Schema:

```
<xs:complexType name="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="ScopedLocalizedTerm_PropertyType " minOccurs="0"/>
      <xs:element name="thesaurusName" type="gmd:CI_Citation_PropertyType" minOccurs="0" maxOccurs=”1”/>
      <xs:element name="linkedResource" type="OnlineResource" minOccurs="0"
           maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
```

Examples

Simple keywords element.

```
<Keywords>
  <keyword>
    <ScopedLocalizedTerm
      xlink:href=http://resource.usgin.org/canonical/urn&#58;USGIN.terms.3463278
      rdf:about="urn&#58;USGIN.terms.3463278">igneous rock</ScopedLocalizedTerm>
  </keyword>
  <keyword>
    <ScopedLocalizedTerm
      xlink:href="http://resource.usgin.org/canonical/urn&#58;USGIN.terms.345578"
      rdf:about ="urn&#58;USGIN.terms.345578">granitoid</ScopedLocalizedTerm>
  </keyword
</Keywords>
```

Keywords on steroids.

```
<Keywords>
  <keyword>
    <ScopedLocalizedTerm
      xlink:href="http://resource.usgin.org/canonical/urn&#58;USGIN.terms.3463278"
      rdf:about="urn&#58;USGIN.terms.3463278">igneous rock</ScopedLocalizedTerm>
  </keyword>

  <keyword>
    <ScopedLocalizedTerm
      xlink:href="http://resource.usgin.org/canonical/urn&#58;USGIN.terms.345578"
      rdf:about ="urn&#58;USGIN.terms.345578">granitoid</ScopedLocalizedTerm>
  </keyword>

  <type>
    <!-- need vocabularyID as base for compound URI for rdf:about to make a
      globally unique identifier -->
    <ScopedLocalizedTerm
  xlink:href="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas
/resources/Codelist/gmxCodelists.xml#MD_KeywordTypeCode"
    vocabularyID="urn&#58;ISO.19139.codelist.MD_KeywordTypeCode"
    rdf:about="theme">theme</ScopedLocalizedTerm>
  </type>

  <type>
    <ScopedLocalizedTerm
      xlink:href="http://resource.usgin.org/canonical/urn&#58;USGIN.KeywordClass.238"
      rdf:about ="urn&#58;USGIN.KeywordClass.238">lithology</ScopedLocalizedTerm>
  </type>

  <thesaurusName>
    <!-- wishful thinking for embedding linked snippets… -->
    <Citation
      xlink:href="http://resource.usgin.org/citation/urn&#58;USGIN.vocabulary.132"
      xlink:show="embed"  xlink:actuate="onLoad"/>
  </thesaurusName>

<!-- extended encoding for linked resource -->
  <linkedResource>
    <linkage>
      <URL>http://service.usgin.org/ontology/</URL>
    </linkage>

    <resourceType>
      <ScopedLocalizedTerm
      xlink:href="http://services.usgin.org/wsdl/urn&#58;USGIN.Service.9608"
      xlink:role="WSDL"
      rdf:about="urn&#58;USGIN.Service.9608">OWL service</ScopedLocalizedTerm>
    </resourceType>

    <resourceName>
      <CharacterString>USGIN lithology ontology</CharacterString>
    </resourceName>

    <linkFunction>
      <ScopedLocalizedTerm
  xlink:href="http://resource.usgin.org/canonical/urn&#58;USGIN.LinkFunction.1532"
      rdf:about="urn&#58;USGIN.LinkFunction.1532">Service Base URL
      </ScopedLocalizedTerm>
    </linkFunction>
  </linkedResource>

  <!-- alternate encoding for online resource using xlink, for simple links -->
  <linkedResource
  xlink:href="http://resource.usgin.org/textExplanation/urn&#58;USGIN.terms.3463278"
    xlink:title="Term definition"/>
  <linkedResource
  xlink:href="http://resource.usgin.org/keywordGuidance/urn&#58;USGIN.terms.3463278"
    xlink:title="Usage guidance"/>
</Keywords>
```

Summary and recommendations

The objective is to develop a standard pattern for using controlled vocabulary concepts in xml documents. The solution seems relatively straightforward, mostly requiring clearer guidance on the usage of attributes, and some conventions about construction of identifiers and means to simplify dereferencing.

USGIN uri encoding

Links

List of registered content types (MIME types). ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/.

References

Linked Data for the Web, http://www.rdfabout.com/intro/?section=8 accessed 2010-02-07

Brickley, Dan, 2000-09-08, Notes on RDF, Xlink as linked information systems: http://www.w3.org/2000/02/rdf-xlink/ accessed 2010-02-07

Booth, David, 2003-01-28, Four uses of a URL: Name, Concept, Web Location and Document instance: accessed at http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm (2010-02-15).

Fielding, R., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., 1999-06, Hypertext Transfer Protocol -- HTTP/1.1: RFC2616, accessed at http://tools.ietf.org/html/rfc2616 2010-02-19.

Mendelsohn, Noah, and Williams, Stuart, 2007-01-02, The use of Metadata in URIs: W3C TAG Finding, accessed at http://www.w3.org/2001/tag/doc/metaDataInURI-31-20070102.html.

Thomson, Henry S., and Rees, Jonathan, 2009-06-22, Dirk and Nadia design a naming scheme or Web naming schemes good practices: W3C Consortium, accessed at http://www.w3.org/2001/tag/doc/justSayHTTP 2010-02-19.

Lewis, Rhys, editor, 2007-05-31, Dereferencing HTTP URIs: W3C Draft Tag Finding, 10 p., Accessed at http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14.html 2010-02-19 (note current version dated 2007-10-04 strikes all content from this document).

Glossary

Controlled Vocabulary

Concept

Dereference: C and C++, an operation that may be applied to a pointer to access the variable or memory location that the pointer points to. The act of retrieving a representation of a resource identified by a URI (

Uniform Resource Locator (URL): A URI that refers to an object accessed with an existing protocol (RFC1630). A URI that provides a means of locating the resource by describing its primary access mechanism (e.g., its network "location") (RFC 3986)

Identifier: An identifier embodies the information required to distinguish what is being identified from all other things within its scope of identification. The terms "identify" and "identifying" refer to the purpose of distinguishing one resource from all other resources, regardless of how that purpose is accomplished (e.g., by name, address, or context). These terms should not be mistaken as an assumption that an identifier defines or embodies the identity of what is referenced, though that may be the case for some identifiers. Nor should it be assumed that a system using URIs will access the resource identified: in many cases, URIs are used to denote resources without any intention that they be accessed. Likewise, the "one" resource identified might not be singular in nature (e.g., a resource might be a named set or a mapping that varies over time). Instances of URIs from any given scheme may have the characteristics of names or locators or both, often depending on the persistence and care in the assignment of identifiers by the naming authority, rather than on any quality of the scheme. (RFC-3986)

URI: an identifier consisting of a sequence of characters matching the syntax rules specified in Section 3 of RFC-3986. URIs have a global scope and are interpreted consistently regardless of context, though the result of that interpretation (dereferencing) may be in relation to the end-user's context. For example, http://localhost/ has the same interpretation for every user of that reference, even though the network interface corresponding to "localhost" may be different for each end-user: interpretation is independent of access. A URI can be further classified as a locator, a name, or both.

Uniform Resource Name: (URN) has been used historically to refer to both URIs under the "urn" scheme [RFC2141], which are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable, and to any other URI with the properties of a name. (RFC-3986)

Information resource: a resource that is a digital artifact, for which the actual resource can be transmitted electronically. Examples – a text document, sound recording, image. Different representations are possible, but these are distinguished only by file formatting and digital encoding variations.

Non-information resource: a resource that is an entity in the world that can not be transmitted electronically. The resource is represented on the World Wide Web by one or more information resources.

Specific representation: a particular instance of an information resource in a specific file format and encoding. This representation may be duplicated many times, but its identity is based on bit-wise digital equivalence of the 'significant content'.

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