MAYO CLINIC
DIVISION OF BIOMEDICAL INFORMATICS

LexGrid Source Mapping Guide

LexGrid Vocabulary Services for caBIG™ (LexBIG)

Version 1.0

Last Modified: September 26, 2008

Authors: Scott Bauer
Craig Stancl

MAYO CLINIC

LexGRID SOURCE MAPPING GUIDE

Document History

Document Location

This document can be found on the GForge site at:

Revision History

Version Number	Revision Date	Author	Summary of Changes
1.0		Scott Bauer Craig Stancl	Initial Draft

Review

Name	Team/Role	Version	Date Reviewed	Reviewer Comments

Related Documents

More information can be found in the following related documents:

Document Name
LexGrid_Loader_Mapping.xls

Approval

Name	Team/Role	Version	Date Reviewed	Signature

TABLE OF CONTENTS

1 Unified Medical Language System

The Unified Medical Language System (UMLS) and Rich Release Format (RRF) files

The UMLS' large medical thesaurus is available as a set of text based, "|' separated files which can be made subset into individual terminologies depending on the user's needs. NCI's MetaThesaurus is also RRF formatted. We map individual terminologies, the entire NCI MetaThesaurus and the UMLS terminology SEMNET into LexGrid Using specific loaders and mappings for each.

Supported Coding Scheme Attributes:

These aren't mapped as categories to a model element. That is, a supported association has an attributeTag column with a corresponding name, but it's context is implied in the name of the supported attribute. For instance, supported associations will have an attributeTag of "association" but that tag corresponds to no element in the model element SupportedAssociation. Instead the context is implied in the name of the element SupportedAssociation.

Preferred Presentation Selection:

Preferred Presentation is determined first by sorting the presentations to include first those in the default language of the Terminology. Following that and given there is more than one presentation in the default language the "most preferred" is determined in the following manner:

Using the "isPref" column, the "TS" and "STT" columns in the MRCONSO RRF file, or a combination of these columns. The MRRANK file overrides these columns.

Preferred Definition Selection:

Definitions in UMLs are not ranked, the first definition found for a concept in the source file MRDEF.RRF is set to preferred.

Special SNOMED adjustments for concept presentation language:

Snomed handles it's language default settings differently than other UMLS terminologies, we hard code it's default language as "en" as a result.

Presentation language is determined by combining the values of SUI, LUI and CUI from MRCONSO and selecting the ATV value from MRSAT where SAB always equals SNOMEDCT and the ATN value is either LANGUAGECODE or SUBSETLANGUAGECODE.

Association Qualifiers for medDRA and others:

MedDRA employs SMQ's or Standardized Medical Queries as a method of classifying portions of this terminology. These are expressed in MRSAT.RRF when the AUI in the METAUI column is replaced by a RUI code. In LexBIG is RUI is identified in the MRREL.RRF source as relationships are loaded and the associated ATN and ATV values from the MRSAT.RRF row are populated as association qualifier name and value.

Hierarchies expressed in source contexts:

Hierarchies in the UMLS are expressed in the MRREL.RRF file as source, target pairs. However source hierarchies may also be expressed in the MRHEIR.RRF file. These context based hierarchies are realized in LexBIG by accessing the MRHEIR source where the HCD column value is populate. When this is the case, as in MESH, the path of AUI's to root from the code in the HCD column is processed as a hierarchy. LexBIG's behavior is as follows:

- Entries in MRHIER that define multiple contexts (HCD field) per CUI will trigger additional tracking within the LexBIG environment.

- Each link is tracked via the corresponding contextual chain(Path To Root field). To do this, we add association qualifiers that tag the association between each participating concept. The qualifier name is 'HCD' and the value will be the HCD field value from the MRHIER file.

- An individual association between two concepts can participate in multiple context chains by assigning additional association qualifiers. A complete flow across the entire chain of links (essentially reconstructing PTR field) can be derived by recursive evaluation of surrounding links that have the same context qualifications. Since each concept can carry multiple text presentations, property qualifiers will be used to track the individual terms used in each context.

- As with associations, multiple qualifiers can be assigned to each text property. Once again, the qualifier name will be 'HCD' and the value will be the HCD field value from the MRHIER file.

- In order to query context-specific relationships, we can first use the API to filter the relationships a concept participates in, then query neighboring nodes to determine the complete context path, and finally map back to specific terms through the registered HCD qualifiers .

2 OBO Mapping

The OBO each remark in the document header will be combined and put into the coding scheme entityDescription.

For example:

```
remark: autogenerated-by:     DAG-Edit version 1.320
remark: saved-by:             mariacos
remark: date:                 Fri Jun 27 09:41:28 EDT 2003
remark: version: $Revision: 1.1 $
```

3 Protege OWL

DatatypeProperty Representation

Owl:

```
<owl:DatatypeProperty rdf:ID="currency">
          <rdfs:domain rdf:resource="#Money"/>
          <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:DatatypeProperty>
```

In LexGrid, a DatatypeProperty is combination of a conceptProperty and Assocation.

Concept Property

```
<lgCon:concept id="Money">
      <lgCommon:entityDescription>Money</lgCommon:entityDescription>
      ….
     <lgCon:conceptProperty propertyId="P0003" propertyName="currency">
        <lgCommon:text>xsd:string</lgCommon:text>
      </lgCon:conceptProperty>
    </lgCon:concept>
```

Association

```
<lgRel:association id="hasDomain" forwardName="hasDomain"
isReflexive="false" isSymmetric="false" isTransitive="true"
reverseName="kindIsDomainOf">
      <lgRel:sourceConcept sourceEntityType="association"
sourceId="currency">
        <lgRel:targetConcept targetEntityType="concept"
targetId="Money"/>
      </lgRel:sourceConcept>

 <lgRel:association id="currency">
      <associationProperty propertyId="P0007"
propertyName="isDatatypeProperty">
        <lgCommon:text>true</lgCommon:text>
      </associationProperty>
      <associationProperty propertyId="P0008"
propertyName="isObjectProperty">
        <lgCommon:text>false</lgCommon:text>
      </associationProperty>
    </lgRel:association>
 <lgRel:association id="hasDatatype" forwardName="hasDatatype">
      <lgRel:sourceConcept sourceEntityType="association"
sourceId="currency">
        <lgRel:targetDataValue dataId="D0001">
          <lgRel:dataValue>string</lgRel:dataValue>
        </lgRel:targetDataValue>
```

Equivalent Class Representation

Owl:

```
<owl:Class rdf:ID="Father">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Person"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasSex"/>
            </owl:onProperty>
            <owl:hasValue rdf:resource="#MaleSex"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#Person"/>
            <owl:onProperty>
              <owl:ObjectProperty rdf:about="#hasChild"/>
            </owl:onProperty>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

In LexGrid, the equivalentClass is represented as an Association.

Association

```
<lgRel:association id="equivalentClass" forwardName="equivalentClass"
isReflexive="true" isSymmetric="true" isTransitive="true"
reverseName="equivalentClass">
  <lgRel:sourceConcept sourceEntityType="concept" sourceId="Father">
        <lgRel:targetConcept targetEntityType="concept"
      targetId="A38"/>
      </lgRel:sourceConcept>
```

Restriction Representation

Owl:

```
<owl:Class rdf:ID="Large-Format">
          <rdfs:subClassOf rdf:resource="#Camera"/>
          <rdfs:subClassOf>
```

```
               <owl:Restriction>
                     <owl:onProperty rdf:resource="#body"/>
                     <owl:allValuesFrom
rdf:resource="#BodyWithNonAdjustableShutterSpeed"/>
               </owl:Restriction>
          </rdfs:subClassOf>
     </owl:Class>
```

In LexGrid, a restriction is a combination of association and qualifier.

Association:

```
<lgRel:association codingSchemeId="p1" id="body" forwardName="body"
isFunctional="false" isReverseFunctional="false" isSymmetric="false"
isTransitive="false">
      <lgRel:sourceConcept sourceCodingScheme="p1"
sourceEntityType="concept" sourceId="Large-Format">
        <lgRel:targetConcept targetEntityType="concept"
targetId="BodyWithNonAdjustableShutterSpeed">
          <lgRel:associationQualification
associationQualifier="owl:allValuesFrom"/>
        </lgRel:targetConcept>
      </lgRel:sourceConcept>
      <associationProperty propertyId="P0021"
propertyName="isDatatypeProperty">
        <lgCommon:text>false</lgCommon:text>
      </associationProperty>
      <associationProperty propertyId="P0022"
propertyName="isObjectProperty">
        <lgCommon:text>true</lgCommon:text>
      </associationProperty>
    </lgRel:association>
```

Additional Examples

Owl:

```
<owl:Class rdf:ID="Father">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Person"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:FunctionalProperty rdf:about="#hasSex"/>
            </owl:onProperty>
            <owl:hasValue rdf:resource="#MaleSex"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:someValuesFrom rdf:resource="#Person"/>
            <owl:onProperty>
              <owl:ObjectProperty rdf:about="#hasChild"/>
            </owl:onProperty>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
```

LexGrid:

```
<lgRel:association id="equivalentClass" forwardName="equivalentClass"
isReflexive="true" isSymmetric="true" isTransitive="true"
reverseName="equivalentClass">
  <lgRel:sourceConcept sourceEntityType="concept" sourceId="Father">
        <lgRel:targetConcept targetEntityType="concept"
      targetId="A38"/>
      </lgRel:sourceConcept>


 <lgRel:association codingSchemeId="" id="hasSex" forwardName="hasSex"
isFunctional="true" isReverseFunctional="false" isSymmetric="false"
isTransitive="false">
    <lgRel:sourceConcept sourceEntityType="concept" sourceId="A38">
        <lgRel:targetConcept targetEntityType="concept"
targetId="MaleSex">
          <lgRel:associationQualification
associationQualifier="owl:hasValue"/>
        </lgRel:targetConcept>

<lgRel:association codingSchemeId="rdfs" id="subClassOf"
forwardName="subClassOf" isFunctional="false" isReflexive="true"
isSymmetric="false" isTransitive="true" reverseName="hasSubClass">
   <lgRel:sourceConcept sourceEntityType="concept" sourceId="A38">
        <lgRel:targetConcept targetEntityType="concept"
targetId="Person"/>
      </lgRel:sourceConcept>

 <lgRel:association codingSchemeId="" id="hasChild"
forwardName="hasChild" isFunctional="false" isReverseFunctional="false"
isSymmetric="false" isTransitive="false">
   <lgRel:sourceConcept sourceEntityType="concept" sourceId="A38">
        <lgRel:targetConcept targetEntityType="concept"
targetId="Person">
          <lgRel:associationQualification
associationQualifier="owl:someValuesFrom"/>
        </lgRel:targetConcept>

<lgCon:concept id="A38" isAnonymous="true">
      <lgCommon:entityDescription>Person and (hasSex has MaleSex) and
(hasChild some Person)</lgCommon:entityDescription>
      <lgCon:presentation propertyId="P0002"
propertyName="textualPresentation" isPreferred="true">
        <lgCommon:text>Person and (hasSex has MaleSex) and (hasChild
some Person)</lgCommon:text>
      </lgCon:presentation>
      <lgCon:conceptProperty propertyId="P0001" propertyName="type">
        <lgCommon:text>owl:intersectionOf</lgCommon:text>
      </lgCon:conceptProperty>
    </lgCon:concept>
```

Property Restriction Representation

Anonymous LexGrid concepts are created for property restrictions (UnionOf, hasValue).

Example 1

Owl:

```
<owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#Hot"/>
          <owl:Class rdf:ID="Medium"/>
          <owl:Class rdf:about="#Mild"/>
        </owl:unionOf>
      </owl:Class>
```

LexGrid:

```
<lgCon:concept id="A17" isAnonymous="true">
      <lgCommon:entityDescription>Hot or Medium or 
Mild</lgCommon:entityDescription>
      <lgCon:presentation propertyId="P0001" 
propertyName="textualPresentation" isPreferred="true">
        <lgCommon:text>Hot or Medium or Mild</lgCommon:text>
      </lgCon:presentation>
      <lgCon:conceptProperty propertyId="P0002" propertyName="isUnion">
        <lgCommon:text>true</lgCommon:text>
      </lgCon:conceptProperty>
      <lgCon:conceptProperty propertyId="P0003" 
propertyName="isIntersection">
        <lgCommon:text>false</lgCommon:text>
      </lgCon:conceptProperty>
      <lgCon:conceptProperty propertyId="P0004" 
propertyName="isEnumeration">
        <lgCommon:text>false</lgCommon:text>
      </lgCon:conceptProperty>
    </lgCon:concept>

Example 2
```

Owl:

```
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasTopping"/>
                <owl:allValuesFrom>
                    <owl:Class>
                        <owl:unionOf rdf:parseType="Collection">
                            <owl:Class rdf:about="#MozzarellaTopping"/>
                            <owl:Class 
rdf:about="#PeperoniSausageTopping"/>
```

```
                            <owl:Class
rdf:about="#JalapenoPepperTopping"/>
                            <owl:Class rdf:about="#TomatoTopping"/>
                            <owl:Class
rdf:about="#HotGreenPepperTopping"/>
                        </owl:unionOf>
                    </owl:Class>
                </owl:allValuesFrom>
            </owl:Restriction>
```

LexGrid:

```
<lgRel:association id="hasTopping" forwardName="hasTopping"
isFunctional="false" isNavigable="true" isReverseFunctional="true"
isSymmetric="false" isTransitive="false">

  <lgRel:sourceEntity sourceCodingScheme="pizza"
sourceEntityType="concept" sourceId="AmericanHot">
        <lgRel:targetEntity targetCodingScheme="pizza"
targetEntityType="concept" targetId="A16">
          <lgRel:associationQualification
associationQualifier="owl:allValuesFrom"/>
        </lgRel:targetEntity>
      </lgRel:sourceEntity>
  </lgRel:association>
```

```
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="#hasTopping"/>
                <owl:allValuesFrom>
                    <owl:Class>
                        <owl:unionOf rdf:parseType="Collection">
                            <owl:Class rdf:about="#MozzarellaTopping"/>
                            <owl:Class
rdf:about="#PeperoniSausageTopping"/>
                            <owl:Class
rdf:about="#JalapenoPepperTopping"/>
                            <owl:Class rdf:about="#TomatoTopping"/>
                            <owl:Class
rdf:about="#HotGreenPepperTopping"/>
                        </owl:unionOf>
                    </owl:Class>
                </owl:allValuesFrom>
            </owl:Restriction>
        </rdfs:subClassOf>
```

```
<lgCon:concept id="A16" isActive="true" isAnonymous="true">
      <lgCommon:entityDescription>MozzarellaTopping or
PeperoniSausageTopping or JalapenoPepperTopping or TomatoTopping or
HotGreenPepperTopping</lgCommon:entityDescription>
      <lgCon:presentation propertyId="P0002"
propertyName="textualPresentation" isPreferred="true">
```

```
        <lgCommon:text>MozzarellaTopping or PeperoniSausageTopping or
JalapenoPepperTopping or TomatoTopping or
HotGreenPepperTopping</lgCommon:text>
      </lgCon:presentation>
      <lgCon:conceptProperty propertyId="P0001" propertyName="type">
        <lgCommon:text>owl:unionOf</lgCommon:text>
      </lgCon:conceptProperty>
    </lgCon:concept>
```

4 NCI OWL

Top-level containers for relations are created, which separate the association types based on the notion of ‘associations’ and ‘roles’ as defined by NCI:

- Associations are “non-inheritable, non-defining relations between concepts”
- Roles are “inheritable relationships”

A LexGrid concept is created for every anonymous class present in the OWL ontology.

If no equivalent class for a concept, it is considered primitive and is indicated by creating a concept property set to ‘true.’

Embedded XML

Property text with embedded XML fragments are identified by by the following identifiers:

qual-name
qual-value
qual

If the extracted tag is one of XML Text identifiers:

Value
term-name
def-definition
go-term

The text of the property is set to the tag value.

If the extracted tag is one of XML Source Name identifiers:

term-source
def-source

A property source is created and the tag value identifies the source.

If the property is a presentation and the extracted tag is XML Representational Form:

term-group

The representational form of the presentation property is set to the tag value.

If the extracted tag is one of DB XRef Prefix:

dbxref.*

A property qualifier is created. The property qualifier id is set to the tag, the value is set to the tag value.

5 HL7 RIM

To build a single coding scheme from the HL7 MS Access database, implementation is similar to how the NCI MetaThesaurus is stored in LexGrid.

For example, here is how entries MTHU021347 and MTHU033458 in ICPC2ICD10ENG (NCI MethThesaurus C1394796) are structured in LexGrid:

Coding Scheme: NCI MetaThesaurus - urn:oid:2.16.840.1.113883.3.26.1.2
Concept Code: C1394796
Entity Description: decompensation; heart, senile
Status: Active
Is Active: true
Is Anonymous: false
Presentation: decompensation; heart, senile
  Property Name: textualPresentation
  Property Id: T-1
  Language: ENG
  Is Preferred: true
  Representational Form: PT
  Source: ICPC2ICD10ENG , **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** MTHU021347
Presentation: heart; decompensation, senile
  Property Name: textualPresentation
  Property Id: T-2
  Language: ENG
  Is Preferred: false
  Representational Form: PT
  Source: ICPC2ICD10ENG , **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** MTHU033458
ConceptProperty: Mental or Behavioral Dysfunction
  Property Name: Semantic_Type
  Property Id: SemType-1

In HL7, code systems, concepts, and designations are in the following tables:

Table: VCS_concept_code_xref

Internal concept identifier	Code system OID	Concept code	Case difference	Status
10011	2.16.840.1.113883.5.55	M	0	A
10011	2.16.840.1.113883.5.55	R	0	A
10013	2.16.840.1.113883.5.55	RQ	0	A
10014	2.16.840.1.113883.5.55	NP	0	A
10015	2.16.840.1.113883.5.55	NR	0	A
10016	2.16.840.1.113883.5.55	RE	0	A
10017	2.16.840.1.113883.5.55	X	0	A
10019	2.16.840.1.113883.5.57	R	0	A

10020	2.16.840.1.113883.5.57	D	0	A
10021	2.16.840.1.113883.5.57	I	0	A
10022	2.16.840.1.113883.5.57	K	0	A
10023	2.16.840.1.113883.5.57	V	0	A
10025	2.16.840.1.113883.5.57	ESA	0	A
10026	2.16.840.1.113883.5.57	ESD	0	A
10027	2.16.840.1.113883.5.57	ESC	0	A
10028	2.16.840.1.113883.5.57	ESAC	0	A

Table: VCS_concept_designation

Internal Id	Designation	seq - for case differences	language	preferredForLanguage
10011	Mandatory	0	en	-1
10011	Required - V2.x	0	en	0

Query of HL7 internal id, concept code and designation:

codeSystemName	Code system OID	Internal concept identifier	Concept code	Designation
HL7ConformanceInclusion	2.16.840.1.113883.5.55	10011	R	Required - V2.x
HL7ConformanceInclusion	2.16.840.1.113883.5.55	10011	M	Mandatory
HL7ConformanceInclusion	2.16.840.1.113883.5.55	10011	M	Required - V2.x
HL7ConformanceInclusion	2.16.840.1.113883.5.55	10011	R	Mandatory

To represent HL7 in LexGrid:

A single coding scheme will be created in LexGrid.

Each **VCS_concept_code_xref.internalId** will be represented as a LexGrid Concept Code.

The LexGrid Concept Code will be generated by the concatination of **VCS_concept_code_xref.internalId** and **VCS_concept_code_xref.conceptCode2** (separated by a colon ':').

Not only the duplicates that exist within coding schemes will be dealt with using the id/mnemonic concatenation but also those duplicates that exist between coding schemes.

A LexGrid Concept Code Presentation Property will be created for each HL7 designation (VCS_concept_designation).

The Presentation Property will include Presentation (HL7 Designation), Source (HL7 codeSystemName) and a Property Qualifier of source-code (HL7 Concept Code).

For example, the following structure represents both HL7 10011 entries in code system 2.16.840.1.113883.5.55:

Coding Scheme: HL7 - urn:oid:2.16.840.1.113883.3.26.1.2
Concept Code: 10011:M
Entity Description: >The message element must appear every time the message is communicated and its value must not be null. This condition is subject to the rules of multiplicity and conditionality. If a non-null default value is defined for the element, a null value may be communicated.
Status: Active
Is Active: true
Is Anonymous: false
Presentation: Mandatory
  Property Name: textualPresentation
  Property Id: T-1
  Language: ENG
  Is Preferred: true
  Representational Form: PT
  Source: HL7ConformanceInclusion , **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** M
Presentation: Required - V2.x
  Property Name: textualPresentation
  Property Id: T-2
  Language: ENG
  Is Preferred: false
  Representational Form: PT
  Source: HL7ConformanceInclusion, **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** M

Coding Scheme: HL7 - urn:oid:2.16.840.1.113883.3.26.1.2
Concept Code: 10011:R
Entity Description: >The message element must appear every time the message is communicated and its value must not be null. This condition is subject to the rules of multiplicity and conditionality. If a non-null default value is defined for the element, a null value may be communicated.
Status: Active
Is Active: true
Is Anonymous: false
Presentation: Mandatory
  Property Name: textualPresentation
  Property Id: T-1
  Language: ENG
  Is Preferred: true
  Representational Form: PT
  Source: HL7ConformanceInclusion , **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** R
Presentation: Required - V2.x
  Property Name: textualPresentation
  Property Id: T-2
  Language: ENG
  Is Preferred: false
  Representational Form: PT
  Source: HL7ConformanceInclusion, **Role:** null, **SubRef:** null
  Property Qualifier Id: source-code , **Property Qualifier Content:** R

Loading the HL7 Rim as a monolithic coding scheme

1. Load coding scheme data as HL7 Rim Metadata from the Model table (rather than the coding scheme data for each HL7 coding scheme).
    a. Mapping of these values will be incomplete:
        i. Mapping proposal:

LexGrid	HL7 RIM
<codingSchemeName>	<modelID>
<formalName>	<name>
<registeredName>	http://www.hl7.org/Library/data-model/RIM *[1]
<defaultLanguage>	en*
<representsVersion>	<versionNumber>
<isNative>	0*
<approximateNumberofConcepts>	Result of count on concept bearing table?
<firstRelease>	MISSING
<modifiedInRelease>	MISSING
<deprecated>	MISSING
<entityDescription>	<description>
<copyright>	MISSING

    b. No URN exists and we may need to consider creating one (see entry for registeredName).

2. Locate and load all mappings (such as supportedAssociations and supportedProperties).
    a. Create a supportedHiearchy with a root node of @ on hasSubtype?

3. Iterate through the code system table rows and get each coding scheme.
    a. Create and persist an “@” node in the database
    b. Prepare an artificial “top node“ for each coding scheme. (Metadata persisted here as concept properties?) This will result in 250 top nodes.
        i. The artificial top nodes will need to have a concept code created for them.
        ii. Attach to “@” the artificial top nodes as a hasSubtype.
        iii. Locate the actual top nodes of each coding scheme by querying the relations table to see if they exist as a target code, if not, they are top nodes so attach them to the artificial top node via hasSubtype.
    c. Translate the RRF source property loads to the EMF world.
        i. Load the concepts ensuring that the coding scheme name is loaded as a “source” property
        ii. Load the relations ensuring that the source and target coding scheme data is loaded with the coding scheme name.
4. Concurrent to this process create an updated “HL7 RIM to LexGrid for NCI” mapping from the current Excel mapping document.

[1] * denotes hard coded string value of some sort.

6 LexGrid Text

The text files that can be imported must use the following formats. Items surrounded by <> are required. Items further surrounded by [] are optional. \t represents a tab - the default delimiter - however other delimiters may be used.

Lines beginning with # are comments.

Format A:

```
<codingSchemeName>\t<codingSchemeId>\t<defaultLanguage>\t<formalName>[\
t<version>][\t<source>][\t<description>][\t<copyright>]
<name1>[\t <description>]
\t <name2>[\t <description>]
\t\t <name3>[\t <description>]
\t\t <name4>[\t <description>]
```

The leading tabs represent hierarchical “hasSubtype” relationship nesting :

(name1 hasSubtype name2 and name2 hasSubtype name3)

Concept Codes will be automatically generated.

If a name is used more than once - it will be assigned the same code.

If a description is used more than once (for a given name) only the first description will be stored.

Format B:

In this format, concept codes can be provided. This is the same as “Format A” with the inclusion of concept codes as part of the input.

```
<code>\t<name>[\t<description>]
```

If the same code occurs twice, the names must match. Description rules same as “Format A.”

Example of Format A

```
#lines starting with "#" are comments

#blank lines are ok

#the first "real" line of the file must be of the following format:
#<codingSchemeName>\t<codingSchemeId>\t<defaultLanguage>\t<formalName>[
\t<version>][\t<source>][\t<description>][\t<copyright>]

colors      1.2.3 en    colors coding scheme     1.0  Someone’s Head
      a simple example coding scheme using colors     This isn't worth
copyrighting

#The rest of the file (for format A) should look like this:

Color Holder of colors
      Red
      Green The color Green
            Light Green foobar
            Dark Green  The color dark green
      Blue
            Red
                  Green The color Green
```

Example of Format B

```
#lines starting with "#" are comments

#blank lines are ok

#the first "real" line of the file must be of the following format:
#<codingSchemeName>\t<codingSchemeId>\t<defaultLanguage>\t<formalName>[
\t<version>][\t<source>][\t<description>][\t<copyright>]

colors2      1.2.4  en     colors coding scheme       1.1    Someone’s Head
       a simple example coding scheme using colors    This isn't worth
copyrighting

#The rest of the file (for format B) should look like this:

1      Color  Holder of colors
       4      Red
       6      Green  The color Green
              7      Light Green
              8      Dark Green
       5      Blue
              8      Dark Green    The color dark green
       6      Green  A different color of green
```