

LexEVS 5.x Design and Architecture Guide 2 - LexGrid Model

Contents of this Page

- [Introduction](#)
- [LexGrid Model Overview](#)
- [Code Systems](#)
- [Concepts](#)
- [Relations](#)

Introduction

This document is a section of the [Design and Architecture Guide](#).

LexGrid Model Overview

The LexGrid Model is Mayo Clinic's proposal for standard storage of controlled vocabularies and ontologies. The LexGrid Model defines how vocabularies should be formatted and represented programmatically, and is intended to be flexible enough to accurately represent a wide variety of vocabularies and other lexically-based resources. The model also defines several different server storage mechanisms and an XML format. This model provides the core representation for all data managed and retrieved through the LexBIG system, and is now rich enough to represent vocabularies provided in numerous source formats such as OWL (NCI Thesaurus) and RRF (NCI MetaThesaurus).

Once the vocabulary information is represented in a standardized format, it becomes possible to build common repositories to store vocabulary content and common programming interfaces and tools to access and manipulate that content. The LexBIG API developed for caBIG® is one such interface, and is described in additional detail in [LexBIG APIs](#).

To view the LexGrid 2010 Logical Model please visit the following: [LexGrid 2010 Model](#)

Following are some of the higher-level objects incorporated into the model definition:

Code Systems

Each service defined to the LexGrid model can encapsulate the definition of one or more vocabularies. Each vocabulary is modeled as an individual code system, known as a *codingScheme*. Each scheme tracks information used to uniquely identify the code system, along with relevant metadata. The collection of all code systems defined to a service is encapsulated by a single *codingSchemes* container.

Concepts

A code system may define zero or more coded concepts, encapsulated within a single container. A concept represents a coded entity (identified in the model as a *concept*) within a particular domain of discourse. Each concept is unique within the code system that defines it. To be valid, a concept must be qualified by at least one designation, represented in the model as a *property*. Each property is an attribute, facet, or some other characteristic that may represent or help define the intended meaning of the encapsulating concept. A concept may be the source for and/or the target of zero or more relationships. Relationships are described in more detail in a following section.

Relations

Each code system may define one or more containers to encapsulate relationships between concepts. Each named relationship (e.g. "hasSubtype" or "hasPart") is represented as an *association* within the LexGrid model. Each relations container must define one or more association. The association definition may also further define the nature of the relationship in terms of transitivity, symmetry, reflexivity, forward and inverse names, etc. Multiple instances of each association can be defined, each of which provide a directed relationship between one source and one or more target concepts.

Source and target concepts may be contained in the same code system as the association or another if explicitly identified. By default, all source and target concepts are resolved from the code system defining the association. The code system can be overridden by each specific association, relation source (*associationInstance*), or relation target (*associationTarget*).