National Cancer Institute

U.S. National Institutes of Health | www.cancer.gov

caBIG Knowledge Center
A part of the Enterprise Support Network

| Home | Knowledge Centers | Discussion Forums | Bugs/Feature Requests | Development Code Repository |

page | discussion | view source | history

LexEVS 5.x Migration OWL loader

LexEVS 5.x Migration Guide > LexEVS 5.x Migration LexEVS model > LexEVS 5.x Migration LexEVS database > LexEVS 5.x Migration EVS API to LexEVS API > LexEVS 5.x Migration OWL loader

Introduction

This document is a section of the Migration Guide.

For more information about the OWL loader, see Loader Guide.

OWL loader enhancements

Substantial changes have been implemented in LexEVS 5.0 during the conversion of the OWL loader. The NCI OWL loader has been decommissioned and replaced with a more generic Protégé OWL loader. All effort has been made to ensure that no previous functionally has been lost during this transition. Priority was given to maintaining existing functionality while improving the OWL loader.

Enhancements and changes made to the OWL loader:

- Improve OWL model footprint by upgrading to latest Protégé (3.4 w/improved support for database streaming)
- Provide ability to enable use the Protégé DB support (Protégé database will serve as cache while we build the LexEVS model from OWL)
- Add support for NCI-based complex props (processing of XML fragments)
- Add support for preferences
- Add support for manifest
- Add support to split role and associations (consider splitat relation container level as done by NCIT loader)
- When resolving IndividualProperties, changed casting from 'OWLNamedClass' to super interface 'RDFSNamedClass'.
- When determining the Entity Id, there were some spots that were using the 'getBrowserText()' method on the 'OWLNamedClass' class. The 'getBrowserText()' was intended to give Protege a nice display string -- but in order to get the id we wanted we want to use the 'getLocalName()' method.
- Now we do not create 'domain' and 'range' associations if there is no target of the association.
- When processing OWLObjectProperties, changed casting from 'OWLObjectProperty' to super interface 'RDFProperty'
- When processing Instances, changed casting from 'OWLNamedClass' to super interface 'RDFSClass', and 'OWLIndividual' to super interface 'RDFResource'.
- When determining the the Entity code during load of an association, we now parse the string based on a colon OR hash symbol.
    For example:
        http://someNamespace.org:C12345 🔗 would resolve to 'C12345'
    and
        http://someNamespace.org#C12345 🔗 would also resolve to 'C12345'
    We used to process only the colon.
- The isDefined() property is now set on created entities.
- Removed the following OWL preferences - dataTypeNameBoolean, associatonNameHasType, and associationNameHasTypeURN.
- Annotation properties are now stored in terms of presentation/comments.
- Manifest supports forward and reverse association names.
- The codedNodeSet restriction added to restrict lucene-based queries.
- RDF local names are used instead of 'textualPresentation' and 'comment' property names.
- Updated SupportedCodingScheme.isImported set to "true" as default.
- The previous NCI Loader and related dependencies have been removed.

- Non-concept entities by EMF EntityService are being handled correctly.

- Memory profiling options 0 and 3 removed from external interfaces.

- Instances are streamed under the enhanced memory profile options.

- Update made to properly store/retrieve the entity type in lucene indexing.

- Update made for use of association code as the 'id' in supported associations are consistent with hierarchy and general API declarations that work with associations (same for GUI interfaces).

- Loader preference "CreateConceptForObjectProp" is added. It controls whether concept entities are created for object properties defined in the OWL source. The default is false.

- Loader preference "DatatypePropSwitch" is added. It controls how data type properties are converted to components of the LexGrid model. If 'association' is specified, each data type property is recorded in LexGrid as an entity-to-entity relationship. If 'conceptProperty' is specified, traditional LexGrid properties are created and assigned directly to new entities. If 'both' is specified, both entity relationships and standard LexGrid entity properties are generated. The default is 'both'.

- Namespace prefixes from the owl source will be registered as supportedNamespace instead of supportedCodingScheme.

- Copyright information is no more hardcoded into the loader. The copyright should be specified in the manifest.

- The Loader will not hardcode the codingschemeName as NCI_Thesaurus. Manifest option has to be used to change it.

- Associations have been distributed among two containers (association and roles)

- Concepts will not have properties "NCI-preferred-term" and "CONCEPT-NAME". How ever, required properties can be introduced by using preferences option "PrioritizedPresentationNames", "PrioritizedDefinitionNames" and "PrioritizedCommentNames".

- Complex properties are not handled by default by the owl loader. Use preference option "ProcessComplexProps" to enable it.

- The restrictions an equivalent class are connected to the parent concept as it was done in NCI-OWL loader. However, if strict owl implementation is required (restrictions an equivalent class not connected to the parent concept) , use the preference option "StrictOWLImplementation"

- Deprecated concepts issue has been resolved by comparing "rdfResource.getRDFType().getName()" with the literal.

- Root node identification: If the preference option "MatchRootName" is specified, the root nodes are identified from it. Otherwise root node is identified from the protege owl api.

- The associations "hasInstance", "hasDomain", "hasRange", "hasDatatype" and "hasDatatypeValue" has been renamed to "instance", "domain", "range", "datatype", "datatypevalue" respectively.

- LexGrid data streaming options have been introduced for effective memory utilizations. Users can choose the memory safe modes based on the requirements.

Categories: VKC Contents | Documentation | LexEVS