

Guide to eagle-i Application ontologies

eagle-i software is predicated upon the principle of ontology layering. Applications are driven by an *application layer ontology* that complements and annotates a *domain ontology*. An application layer ontology encapsulates the specifics of the software behavior and is not meant to be reusable in and of itself, in contrast to a domain ontology. The domain ontology used in the current eagle-i stack is the [ERO \(eagle-i resource ontology\)](#). The layering principle is applicable to other domain ontologies.

[picture]

The runtime model that represents the ontology to the applications is built by loading the domain and application ontologies and interpreting the annotations in different ways. The memory model is organized as a core model fully stored in memory and a collection of referenced taxonomy models stored in Solr indexes for rapid access.

File Structure

- eagle-i-app-def.owl contains axioms that define the basic annotation vocabulary (classes and properties).
- eagle-i-core-app.owl serves as a root for the software loading mechanism. It provides the chain of imports - if an owl file is reachable directly or indirectly from this file via an owl:import statement, it will be loaded as part of the core model. It is a configurable parameter in the software.
- Files named *-app.owl contain the axioms that annotate the domain ontologies. There is typically one app file per domain owl file (e.g. mesh-app.owl -> mesh.owl).
 - Note that eagle-i-core-app.owl serves this purpose for the ero.owl file. Should eventually be renamed to ero-app.owl for consistency
 - (describe mechanism for loading referenced taxonomy files into solr)

Requirements

Every app file must declare an ontology IRI (details and examples to come)

Every app file must declare an ontology version (details and examples to come)

Every app file must import its corresponding domain owl file

Summary of app ontology (eagle-i-app-def.owl)

Namespace

all of the classes and properties below have the namespace:

<http://eagle-i.org/ont/app/1.0/>

Classes

ClassGroup
PropertyGroup

Object properties

These behave as annotation properties, i.e. they appear in triples where the subject is an owl class or an owl property. However their predicate is always an individual of type ClassGroup or PropertyGroup, hence they are declared as object properties.

inClassGroup => range is ClassGroup

inPropertyGroup => range is PropertyGroup

Individuals of type ClassGroup

	Description	Usage in software
ClassGroup_DataModelExclude	classes that are not to be included in the eagle-i model. For example, upper ontology classes.	When the memory model is built, these classes are ignored
ClassGroup_InstanceCreate	classes of which instances are created in the eagle-i repository. In other words, the only asserted types in the eagle-i repository are classes that belong to this group.	Appear under "Resource Types" in the ontology browser. Drive the creation of pull down lists in SWEET forms

ClassGroup_PrimaryResourceType	classes that are considered an eagle-i primary type - they represent biomedical research resources (as opposed to ancillary types needed for describing biomedical research resources). Note that all primary types must be in the InstanceCreate group, but some classes in the instanceCreate group are not considered primary by the eagle-i software because they do not represent research resources (e.g. Document, Person, Organization)	Drives the SWEET left bar Drives (mostly) the Search left bar (exception is Core Laboratory) Drives Search categories in autosuggest
ClassGroup_EmbeddedResourceType	classes whose instances only have significance when associated to an instance of another "containing" class (e.g. construct insert -> construct). In an eagle-i repository, instances of embedded classes are never shared among instances of containing classes there exist a one to one correspondance embedded instance -> containing instance	Appear under "Embedded Types" in the glossary. Appear in SWEET forms as a sub-form of the main form
ClassGroup_NoAssertedInstances	If a class belongs to this group, it will never have asserted instances in the eagle-i repository - these are typically logical definitions that should only be inferred	The SWEET prevents users from directly assigning a one of these types when creating a resource
ClassGroup_ReferencedTaxonomy	Classes that are only used for annotating a resource (as opposed to having instances created). Note that the data is produced with these classes used as instances - in other words, triples exist that have the class URI as a predicate (punning)	Appear under "Referenced Taxonomies" in the ontology browser. Drive the "annotation" pull down hierarchies in SWEET forms

Individuals of type PropertyGroup

PropertyGroup_AdminData
PropertyGroup_ContactInformation
PropertyGroup_DataModelExclude
PropertyGroup_SearchFacet
PropertyGroup_ResourceProvider
PropertyGroup_PrimaryProperty
PropertyGroup_SecondaryProperty
PropertyGroup_TertiaryProperty

Annotation properties

domainConstraint
preferredDefinition
preferredLabel
rangeConstraint
referencedTaxonomyIRI

Usage