ONSTR Doc03
Snezana Nikolic

ONSTR Term Deprecation Policy

Created: 08-18-2011 (SN)
Last updated: 08-24-2011 (SN)

0. Summary

This document describes ONSTR practices used in deprecating (obsoleting) terms. Deprecation of a term (class/instance, object property and/or annotation property) is a multiple step procedure following the ONSTR team decision to obsolete a term based on the term being detected as redundant or the definition of the term would need to be substantially changed, i.e. the term's meaning would change.
The ONSTR deprecation policy, is based and motivated by the OBI deprecation policy published at: http://obi-ontology.org/page/OBIDeprecationPolicy.

The home of all obsoleted classes and object properties in ONSTR is a file ONSTR_obsolete.owl. Class labeled as *ONSTR obsolete class* is a designated super class for all the retired ONSTR classes. A super property of all deprecated object properties is *ONSTR obsolete property*.
ONSTR_obsolete.owl is imported into the ONSTR.owl via owl:imports.

The deprecation of a term generally involves the three steps:
1) In the ONSTR.owl, locate the code for the term to be obsoleted.
2) Move (cut-paste) the code from ONSTR.owl into the ONSTR_obsolete.owl
3) Make necessary annotation properties modifications to the moved code and update the references (if any) in the ONSTR.owl for the term that has been deprecated.

1. ONSTR Class, Instance and Object Property Deprecation

Deprecation of a class and/or a class instance involves the following steps:

1. Locate the code of the to-be-deprecated class/instance in a ONSTR.owl file.

2. If to-be-deprecated-class is a parent class (i.e. subsumes at least one sub-class), the steps described below need to be taken in order to avoid creating mess in the ontology.

i) If class-to-be-deprecated has at least one sub-class, and if they children sub-classes need not to get deprecated, then these sub-classes need to become children of another parent-class. In other words, for each child class, a newly assigned parent-class URI needs to be updated in text editor to

reflect the superclass re-assignment (<rdfs:subClassOf rdf:resource=”NEW-parent-classURI”/>). If working in Protégé, that particular child class needs to be re-assigned (moved into) a new-parent-class.
ii) If class to-be-deprecated does not have sub-classes, then this step can be omitted.
3. Remove all the class restrictions for to-be-deprecated-class.

4. Update all the references to exclude to-be-deprecated-class within the entire ontology.
5. Move (cut-paste) the code for the to-be-deprecated class into ONSTR_obsolete.owl
i) Put it under the mother class of all obsolete classes *ONSTR obsolete class*.
ii) Update the term’s curation status*: Curation status: Obsolete - XXXX* stating the appropriate deprecation reason in the text of annotation property value.
iii) Add the owl:deprecated annotation:

```
<owl:deprecated rdf:datatype="&xsd;boolean">true</owl:deprecated>
```

iv) Add rdfs:comment stating what was the previous hierarchical position of the just deprecated class, e.g. “*Previous position snap:MaterialObject.”*

Below is the example of the code for the obsoleted class:

<SN: Here need to add the code example for deprecated class, as soon as we have one.>

In the case obsoleted class instances (i.e. individuals) rdf:type annotation property should be added in order to place the obsoleted individuals under ONSTR obsolete class:

```
  <owl:NamedIndividual rdf:about="&ONSTR_obsolete;ONSTR_XXXX">
    <rdf:type rdf:resource="&ONSTR_obsolete;ONSTR_XXXXX"/>
    <rdfs:label rdf:datatype="&xsd;string">obsolete instance name</rdfs:label>
  </owl:NamedIndividual>
```

<SN: Need to double check this example.>

1.1 ONSTR Object Property Deprecation

Deprecation steps described above that apply to classes and instances also apply to deprecating object properties. The only major difference between the deprecating classes/instances and, object properties is in a superclass statement which must be updated according to the ontology term type. That is for obsoleted object properties the parent superProperty should be *ONSTR_XXXX ONSTR obsolete property.* The rest of the step to be taken is the rest as described above.

<SN: Here need to add the code example for deprecated object property.>