

Simple Darwin Core based on SKOS(XL)

Thomas Bandholtz

thomas.bandholtz@innog.com

Editor's draft 0.01

2010-08-13

1 Using SKOS without any extension

In this first example we need only two namespaces:

```
@prefix ex: <http://data.example.com/species/>
@prefix skos: <http://www.w3.org/2004/02/skos/core#>
```

Prefix `ex:` is the local namespace of my species taxonomy, prefix `skos:` is the standard namespace of the W3C SKOS recommendation. As we will not use any SKOS extension, there is no namespace for any local RDF Schema.

First we set up a concept scheme.

```
ex:taxonConcepts a skos:ConceptScheme .
```

Would be good practice to add some naming and annotation properties here ... skip that for now.
Describe "Puma concolor" and "Puma concolor cabreræ" in pure SKOS as far as possible.

```
ex:_4711 a skos:Concept ;
    skos:inScheme ex:taxonConcepts ;
    skos:prefLabel "Puma concolor (Linnaeus, 1771) sec. Brown"@la ;
    skos:prefLabel "cogar"@en-US ;
    skos:prefLabel "puma"@fr ;
    skos:prefLabel "Puma"@de ;
    skos:altLabel "panther"@en-US ;
    skos:altLabel "couguar"@fr ;
    skos:altLabel "Silberlöwe"@de ;
    skos:altLabel "Berglöwe"@de ;
    skos:altLabel "Kuguar"@de ;
    skos:narrower ex:_4712 .

ex:_4712 a skos:Concept ;
    skos:inScheme ex:taxonConcepts ;
    skos:prefLabel "Puma concolor cabreræ"@la ;
    skos:broader ex:_4711.
```

Some notes:

ex:_4711 is short for http://data.example.com/species/_4711. If this is a working http URI, it is globally unique as long as "_4711" is unique within my namespace. We could also use a GUID instead of "_4711", no problem, but from the Linked Data point of view we do not need it: the taxon is not identified by the number, but by the whole URI.

Both taxons are individuals of **skos:Concept**. In this approach we avoid any sub-classing just to find out how far we can go with SKOS alone. There is some indication of a specific kind of a Concept by the assertion **skos:inScheme ex:taxonConcepts**.

skos:prefLabel is used with language tags. The one tagged with @la is what DWC would call a „*scientificName*“. Certainly this is semantically not really equivalent, but it is the closest match as long as we avoid any SKOS extension. The other ones are preferred names in some living languages.

skos:altLabel comes close to what DWC calls “*vernacularName*”.

As both label types are `rdfs:subPropertyOf` `rdfs:label`, we do not need any further labeling here.

skos:narrower and **skos:broader** express the taxon hierarchy. The two property definitions are defined `owl:inverseOf` each other, so we could only make one of the two explicit and leave the rest to the reasoner.

2 What about TaxonRank?

So far, we are missing the TaxonRank information. We could annotate the rank through a `skos:note` for each individual.

```
# not recommended:
ex:_4711 skos:note „species“@en.
```

But this is just some literal value. DWC recommends an enumeration of taxon ranks. Using native SKOS, we would implement this as a second concept scheme¹.

```
ex:taxonRanks a skos:ConceptScheme .

ex:species a skos:Concept ;
  skos:inScheme ex:taxonRanks .

ex:subSpecies a skos:Concept ;
  skos:inScheme ex:taxonRanks ;
  skos:broader ex:species .
```

(Dropped `prefLabel` etc. for these concepts). Now we are looking for a relation between the taxon and the rank. SKOS provides a set of “mapping properties”. Any of those indicate that two concepts from different `ConceptSchemes` have a more or less similar meaning. We do not want to express similarity, but some kind of classification. Here I see the need of a SKOS extension. So we may declare a new property which might be `subPropertyOf` the generic `skos:mappingRelation`:

```
@prefix dwc: <http://rs.tdwg.org/DarwinCore>
dwc:taxonRank rdfs:subPropertyOf skos:mappingRelation ;
```

So we can add:

```
ex:_4711 dwc:taxonRank ex:species.
ex:_4712 dwc:taxonRank ex:subSpecies.
```

3 Adding simple name types

The DWC properties *scientificName* and *vernacularName* are more specific than `skos:prefLabel` and `skos:altLabel`.

¹ Some people might argue we should use a set `skos:Collection` in this case. But using `skos:Collection`, we cannot express the hierarchy of the taxon ranks.

If we want to do so, we can add

```
dwc:scientificName rdfs:subPropertyOf skos:prefLabel.  
dwc:vernacularName rdfs:subPropertyOf skos:altLabel.
```

And exchange the respective statements in the first example:

```
ex:_4711 dwc:scientificName "Puma concolor (Linnaeus, 1771) sec. Brown" ;  
ex:_4711 dwc:vernacularName "panther"@en-US ;  
etc.
```

I dropped the @la language tag in this case, as a scientific name is a more specific thing than a latin name (?).

4 TaxonName using skosxl

There is a more complex notion of TaxonName in DWC, and you can express this with SKOSXL quite well. In SKOSXL, a label is not a literal, but a class instance.

```
ex:pumaConcolor a skosxl:Label ;  
    skosxl:literalForm "Puma concolor (Linnaeus, 1771) sec. Brown"@la ;  
    dwc:genusPart "Puma"@la ;  
    dwc:specificEpithet "concolor"@la ;  
    dwc:authorShip "Linnaeus, 1771" ;  
    dwc:year 1771 ;  
    dwc:hasBasionym ex:felisConcolor ;  
    rdfs:seeAlso http://lsid.tdwg.org/urn:lsid:zoobank.org:act:6FFAFC2C-D46B-4959-BA03-C38477B9DFF1 .
```

```
ex:felisConcolor a skosxl:Label ;  
    skosxl:literalForm "Felis concolor Linnaeus, 1771 sec. Brown"@la ;  
    dwc:genusPart "Felis"@la ;  
    dwc:specificEpithet "concolor"@la ;  
    dwc:authorShip "Linnaeus, 1771" ;  
    dwc:year 1771 .
```

In this case we need to modify the definition of dwc:scientificName and add some more extensions:

```
@prefix skosxl: <http://www.w3.org/2008/05/skos-xl#>  
  
dwc:scientificName rdfs:subPropertyOf skosxl:prefLabel.  
  
dwc:genusPart a rdfs:DatatypeProperty ;  
    rdfs:domain skosxl:Label ;  
    rdfs:range xsd:string .  
  
dwc:hasBasionym rdfs:subPropertyOf skosxl:labelRelation .  
etc.
```

done ;-))