OpenBiodiv RDF Guide

This is the Open Biodiversity Knowledge Management System (OpenBiodiv, formerly known as OBKMS) RDF Guide. The guide is intended to explain to humans and to define for computers the data model of OpenBiodiv and aid its users in generating OpenBiodiv-compatible RDF and in creating useful SPARQL queries or other useful extensions.

This guide is a literate programming document. Literate programming is the act of including source code within documentation. In usual software development practice the reverse holds true. By virtual of this programming paradigm, in this document the formal description of the data model, i.e. the RDF statements that form the ontology and the vocabularies, are found within the document itself and are extracted from it with the program `noweb`. `noweb` can be easily obtained for GNU Linux.

Introduction

Motivation. The raison d'être of the OpenBiodiv Data Model is to enable the operation of a semantic database as part of OpenBiodiv. The data model consists of:

1. A formal computer ontology expressed as RDF, called from here on OpenBiodiv Ontology, introducing the entities that our knowledge base holds and giving axioms that restrict the ways in which they can be combined.
2. Formal vocabularies specified in RDF for particular application areas.
3. Natural language descriptions of the meaning (semantics) of the concepts from (1) and (2) in our conceptualization of the universe of discourse.
4. Examples and recommendations that illustrate and describe the intended model to human users as the formal ontology necessarily will be more lax than the intended model.

Viewing the data model from another angle it

(a) describes a view of the universe of discourse (biodiversity information), which we call conceptualization, and

(b) introduces a formal way to store biodiversity information in a database.

We do not believe other data providers ought to use the same formal way to store biodiversity information in their databases, as they might be using a different database application, or even paradigm. However, we do believe that should information exchange between OpenBiodiv and these other data providers occur, biodiversity information ought to at least follow the same conceptual model presented herein.

For a discussion see Specification of Conceptualization, as well as the article by Guarino et al. (2009).

Def. (OpenBiodiv Ontology): In the following code-chunk that will be extracted by `notangle` (you can use the `Makefile` for this purpose), the top-level structure of the ontology is defined:

```
<<OpenBiodiv Ontology>>=

<<Prefixes>>
<<Ontology Metadata>>
<<Model of the Publishing Domain>>
<<Model of Biological Systematics>>
<<Vocabulary of Taxonomic Statuses>>
<<Vocabulary of RCC5 Terms>>
<<Borrowed Parts from External Ontology>>
@
```

Def. (Ontology Metadata):

```
<<Ontology Metadata>>=

: rdf:type owl:Ontology ;
  owl:versionInfo "0.2" ;
  rdfs:comment "Open Biodiversity Knowledge Management System Core Ontology" ;
  dc:title "OpenBiodiv Core Ontology" ;
  dc:subject "OpenBiodiv Core Ontology" ;
  rdfs:label "OpenBiodiv Core Ontology" ;
  dc:creator "Viktor Senderov, Terry Catapano, Kiril Simov, Lyubomir Penev" ;
  dc:rights "CCBY" ;
  owl:imports <http://phylodiversity.net/dsw/dsw.rdf> ;
  owl:imports <http://www.essepuntato.it/2008/12/pattern> ;
  owl:imports <http://purl.org/spar/fabio/> .

@
```

TODO: Authors list needs to be emended.

Note: We've just defined our *root chunk*, `Ontology`. In the `noweb` way of doing literate programming, we write our source in chunks. Each chunk has a name that is found between the `@<<` (TODO: not sure how to escape this character) and `>>` and ends in `@`. Chunks can contain other chunks and thus the writing of the source code becomes hierarchical and non-linear. In the root chunk, we've listed other chunks that we'll introduce later and some verbatim code. In order to create the ontology we use the `notangle` command from `noweb`.

Command to extract the Core Ontology.

```
notangle -R"OpenBiodiv Ontology" RDF_Guide.md > OpenbBodiv.ttl
```

Examples. This document also contains some examples.

```
<<Examples>>=

 #' These are the examples for the OpenBiodiv data model.
@
```

Command to build the examples.

```
notangle -RExamples RDF_Guide.md > Examples.ttl
```

TODO: check for prefix consistency for all imported ontologies.

Incorporated external ontologies. Our data model is a natural extension of existing data models. Therefore, we incorporate several external ontologies or parts of existing ontologies into ours. We try to include these ontologies via `owl:imports`. Where a URL does not resolve, or we want to import only a specific subset of an ontology or a

specific version, we directly introduce the needed RDF into our model in the code-chunk `Borrowed Parts from External Ontologies`. In addition to that we've downloaded the RDF for everything that we've borrowed in the `imports` sub-subdirectory in case the URL's become unavailable in the future. There is a catalog of this directory under Catalog of imported ontologies, which is, however, still a work in progress.

Prefixes. In OpenBiodiv prefixes are stored in a YAML configuration file called

`prefix_db.yml`

The following Turtle code can be extracted from the prefix database with `obkms::prefix_ttl()` command:

```
<<Prefixes>>=

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix pensoft: <http://id.pensoft.net/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix pro: <http://purl.org/spar/pro/> .
@prefix scoro: <http://purl.org/spar/scoro/> .
@prefix ti: <http://www.ontologydesignpatterns.org/cp/owl/timeinterval.owl#> .
@prefix tvc: <http://www.essepuntato.it/2012/04/tvc/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix fabio: <http://purl.org/spar/fabio/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix frbr: <http://purl.org/vocab/frbr/core#> .
@prefix prism: <http://prismstandard.org/namespaces/basic/2.0> .
@prefix doco: <http://purl.org/spar/doco/> .
@prefix po: <http://www.essepuntato.it/2008/12/pattern#> .
@prefix co: <http://purl.org/co/> .
@prefix trt: <http://plazi.org/treatment#> .
@prefix c4o: <http://purl.org/spar/c4o/> .
@prefix dwciri: <http://rs.tdwg.org/dwc/iri/> .
@prefix nomen: <http://www.semanticweb.org/dmitriev/ontologies/2013/8/untitled-ontology-6#> .
@prefix dwc: <http://rs.tdwg.org/dwc/terms/> .
@prefix sro: <http://salt.semanticauthoring.org/ontologies/sro#> .
@prefix deo: <http://purl.org/spar/deo/> .
@prefix pext: <http://proton.semanticweb.org/protonue#> .
@prefix ptop: <http://proton.semanticweb.org/protont#> .
@prefix pkm: <http://proton.semanticweb.org/protonkm#> .
@prefix : <http://openbiodiv.net/> .
@
```

Types of entities that OpenBiodiv manages. There are two ways to look at the types of entities that OpenBiodiv manages. The first way is to look at the application domain. OpenBiodiv's application domain is the semantic publishing of taxonomic, systematic, biodiversity, genomic, ecologic, and related information. Therefore, the entities that OpenBiodiv manages are separated across these domains.

Another way to look at the entities that OpenBiodiv manages is the structural way. As the main sources of information for OpenBiodiv are scientific articles, we can separate the entities that are extracted in entities that are structural parts of the articles such as articles, paragraphs, sections, tables, figures, etc., and entities that are talked about -- the actual (domain-specific) information contained in the articles.

Both ways of looking at the entities are compatible with each other. The following information should probably go into "OpenBiodiv Extension and Linking Guide" but for the sake of completeness we will present it here, as well.

Entity extraction from taxonomic articles can happen in two phases:

1. Conversion of XML elements into RDF triples following the structure of the XML. In this stage, key XML elements are transformed into bibliographic elements. E.g., XML elements denoting sections become `doco:Section`'s, figures become `doco:Figure`'s, etc. Here no information extraction is taking place and no additional semantics are added. This step can be completed linearly without any external lookups. Key here is that the XML-hierachy is preserved in RDF via, for example, `po:contains`.

2. Named entity recognition, coreferencing, and named entity identification. During this step, non-structural entities are extracted from the informa[illegible] present in the structural elements in the form of text or attributes. The details of how this done are beyond the scope of this guide but it is important to note that an attempt is made to coreference -- i.e. match multiple bibliographic elements to the same non- bibliographic entity, if they do in fact refer to the same entity in the sense of Frege's semiotic trian[illegible] Another attempt is made to properly manage identifiers. This stage cannot be completed without external look-ups.

Note on Capitalization. Our ontology strives be a formal specification of a conceptualization. In our mental model we have some concepts of some thing[illegible] hen we talk about these concepts in the abstract, we will make use of Capitalization. For example, we say Thing for the top-level concept, and we say Treatment when we refer to the concept (introduced later) of a taxon circumscription. We also have concepts for relations (in our conceptualization only binary relations are allowed). To denote these relations in the abstract we use verbal phrases and we might or might not use quotes (we will use quotes only if it adds to the clarity of exposition). For example, Treatment is a Thing as opposed to Treatment "is a" Thing. We also have individual instances of these concepts. To refer to those we might use improper or proper nouns or phrases wherever appropriate. For example, "the treatment on page 5," or "a treatment," or "John."

When we formally define a concept in OWL and issue an URI to it, we shall refer to the URI, as we refer to all URI's in the text with `typewriter font` . URI's of classes and vocabularies will be in `MajorCamelCase` . URI's of relationships will be in `minorCamelCase` . URI's of individuals `will-be-hyphenated` . This seems to generally in accordance with WWW practice.

Examples

```
<<Examples>>=
<<Prefixes>>
@
```

RDF Model

The Publishing Domain

The publishing domain is described in our model using the Semantic Publishing and Referencing Ontologies, a.k.a. SPAR Ontologies. We do import several of these ontologies (please consult the paragraph "Incorporated external ontologies"). Refer to the documentation on the SPAR Ontologies' site for an exhaustive treatment.

In the rest of this section we describe the modeling of entities in the publishing domain that are *not found* in the SPAR ontologies. The central new class in OpenBiodiv not found in SPAR is the `trt:Treatment` class, borrowed from the Treatment Ontologies.

Changes to SPAR

We have mentioned before that when we extract bibliographic elements from the XML, we make use of the `po:contains` SPAR property. For example, an article can `po:contain` a secion and this section can `po:contain` another (sub-)section. In our view, this means that also the article contains the (sub-)section. Thefore we define `po:contains` as a transitive property.

Def. ('contains'):

```
<<Model of the Publishing Domain>>=

po:contains rdf:type owl:TransitiveProperty .
@
```

Article Metadata

The main objects of information extraction and retrieval of OpenBiodiv in the first stage of its development are scientific journal articles from the journals Biodiversity Data Journal and ZooKeys and other Pensoft journals. We model the bibliographic objects around Journal Article, such as Publisher, and Journal using SPAR.

Example:

```
<<Examples>>=

:biodiversity-data-journal rdf:type fabio:Journal ;
  skos:prefLabel "Biodiversity Data Journal" ;
  skos:altLabel  "BDJ" ;
  <http://prismstandard.org/namespaces/basic/2.0/issn>     "1314-2836" ;
  <http://prismstandard.org/namespaces/basic/2.0/eIssn>    "1314-2828" ;
  dcterms:publisher "Pensoft Publishers" ;
  frbr:part <http://dx.doi.org/10.3897/BDJ.4.e10095> .

<http://dx.doi.org/10.3897/BDJ.4.e10095> a fabio:JournalArticle ;
  skos:prefLabel "10.3897/BDJ.4.e10095" ;
  prism:doi "10.3897/BDJ.4.e10095" ;
  fabio:hasPublicationYear "2016"^^xsd:gYear ;
  dcterms:title "A new spider species, Heser stoevi sp. nov., from Turkmenistan (Araneae: Gnaphosidae)"@en .

:pensoft-publishers rdf:type foaf:Agent ;
  skos:prefLabel "Pensoft Publishers" ;
  pro:holdsRoleInTime :pensoft-publishes-bdj .

:pensoft-publishes-bdj rdf:type pro:RoleInTime ;
  pro:relatesToDocument :biodiversity-data-journal .
@
```

TODO: keywords

Note that in this example `:biodiversity-data-journal` is non-structural entity, as it doesn't denote part of the manuscript, but rather something external, i.e. a journal. This means that creating it, requires of the step of named entity identification.

Taxonomic Treatment

See Plazi for a theoretical discussion of Treatment.

Def. (Treatment): *Taxonomic Treatment, or simply Treatment, is a rhetorical element of a taxonomic publication:*

```
<<Model of the Publishing Domain>>=

trt:Treatment a owl:Class ;
  rdfs:label "Taxonomic Treatment"@en ;
  rdfs:label "Taxonomische Abhandlung"@de ;
  rdfs:label "Таксономично пояснение"@bg ;
  rdfs:comment "A taxonomic treatment, or simply a treatment, is a
                rhetorical element of a taxonomic publication, i.e. a
                specialized section, where taxon circumscription
                takes place."@en ;
  rdfs:comment "Eine taxonomische Abhandlung, oder nur Abhandlung, ist
                ein rhetorisches Element eines wissenschaftlichen
                taxomischen Artikels, d.h. ein spezialisierter Abschnitt,
                wo die Umschreibung eines taxonomischen Konzeptes
                stattfindet."@de ;
  rdfs:comment "Таксономично пояснение или само Пояснение е риторчна част
                от таксономичната статия, където се случва описанието
                на дадена таксономична концепция."@bg ;
  rdfs:subClassOf deo:DiscourseElement .
@
```

Thus, Treatment is defined akin to Introduction, Methods, etc. from DEO.

Example: In this example, we show how to instantiate a treatment:

```
<<Examples>>=

:heser-stoevi-treatment
  a doco:Section, trt:Treatment .
@
```

Note that we type `:treatment` both as `trt:Treatment` (i.e. the rhetorical element Treatment) and as s `doco:Section` because we view this particular treatment to also be a structural section of the document.

Example: In this example we show how different sub-article elements such as treatments are linked via the use of the `po:contains` :

```
<<Examples>>=

<http://dx.doi.org/10.3897/BDJ.4.e10095> po:contains :heser-stoevi-treatment .
@
```

Taxonomic Nomenclature Section

Nomenclature is a special subsection of Treatment where nomenclatural acts are published. We define it similar to Treatment, but proper modeling entails that for each Nomenclature there ought to be a Treatment that contains it.

Def. (Nomenclature): Nomenclature is a specialized section of a taxonomic publication, a subsection of Treatment, where nomenclatural acts take place. 

```
<<Model of the Publishing Domain>>=

trt:Nomenclature a owl:Class ;
  rdfs:subClassOf deo:DiscourseElement ,
                  [ rdf:type owl:Restriction ;
                    owl:onProperty po:isContainedBy ;
                    owl:someValuesFrom trt:Treatment ] ;
  rdfs:label "Taxonomic Nomenclature Section"@en ;
  rdfs:comment "A taxonomic nomenclature section, or simply a nomenclature,
                is a rhetorical element of a taxonomic publication, i.e. a
                specialized section, where nomenclatural acts are published."@en .

trt:NomenclatureHeading a owl:Class ;
  rdfs:subClassOf deo:DiscourseElement ,
                  [ rdf:type owl:Restriction ;
                    owl:onProperty po:isContainedBy ;
                    owl:someValuesFrom trt:Nomenclature ] ;
                  rdfs:label "Treatment Title"@en ;
  rdfs:comment "Inside the taxonomic nomenclature section, we have the treatment title."@en .

trt:NomenclatureCitationList a owl:Class ;
  rdfs:subClassOf deo:DiscourseElement ,
                  [ rdf:type owl:Restriction ;
                    owl:onProperty po:isContainedBy ;
                    owl:someValuesFrom trt:Nomenclature ] ;
                  rdfs:label "Taxonomic Nomenclature Citation List"@en ;
  rdfs:comment "Inside the taxonomic nomenclature section, we have a list
                of citations."@en .
@
```

Example: In this example, we show how to define a nomenclature section:

```
<<Examples>>=

:heser-stoevi-treatment
  po:contains :heser-stoevi-nomenclature .
```

```
:heser-stoevi-nomenclature a doco:Section, trt:Nomenclature ;
  po:contains :heser-stoevi-nomenclature-heading .

:heser-stoevi-nomenclature-heading a trt:NomenclatureHeading ;
  cnt:chars
  "Heser stoevi urn:lsid:zoobank.org:act:E4D7D5A0-D649-4F5E-9360-D0488D73EEE8 Deltshev sp. n." .
@
```

TODO: All the other subsections of trt:Treatment, Description, etc.

Taxonomic Name Usage

In the text of taxonomic articles we find strings like "*Heser stoevi* Deltschev, sp. n.". In our conceptualization these are called *taxonomic name usages* (TNU's) as they refer to published scientific names from the domain of biological systematics. The taxonomic name usage consists of three parts:

1. One or more words identifying the taxon (these can be Latinized or take the form of an identifier).
2. The name-and-year of the author(s) of the taxon.
3. Taxonomic name status containing information about the type of the taxonomic name usage.

In the example, "*Heser stoevi*" is the binomial Latinized species name, "Deltschev" is the name of the person who described the taxon and "sp. n." bears taxonomic (and nomenclatural) information indicating that this is a species new to science.

Modeling-wise, we consider TNU's to be specialized instances of `pext:Mention` from the PROTON Extensions module. Furthermore, we link the TNU's to the scientific name they are symbolizing via `pkm:mentions`.

Def. (Taxonomic Name Usage): *A taxonomic name usage is the mentioning of a biological taxonomic name or taxon concept label (see later) in a text, together with possibly a taxonomic status, bearing further information about the name:*

```
<<Model of the Publishing Domain>>=

:TaxonomicNameUsage rdf:type owl:Class ;
  rdfs:subClassOf  pext:Mention ;
  rdfs:comment "A taxonomic name usage is the mentioning of a
biological taxonomic name or taxon concept label in a text."@en ;
  rdfs:label "Taxonomic Name Usage"@en .

dwciri:taxonomicStatus rdf:type owl:ObjectProperty ;
  rdfs:label "taxonomic status"@en ;
  rdfs:comment "the IRI version of the DwC term taxonmic status" .
@
```

Note: In the logic of our algorithms, it is very important that TNU's are dated with `dc:date`.

Example: In the following example, we express in RDF a TNU that is in the nomenclature heading of a treatment (treatment title). Structurally, the TNU is connected to the containing section via `po:contains`; `cnt:chars` is used to dump the full string of the usage and DwC properties are used to encode more granular information in addition to the dump.

In the second step of RDF-ization, we use `dwciri` properties to link the TNU to semantic entities. `dwciri:taxonomicStatus` is used to link the TNU to an item in the OpenBiodiv Taxonomic Status Vocabulary. `:scientificName` is used to link the TNU to the IRI of the name that the TNU is mentioning. Note, we have introduced `:scientificName` as a sub-property of `pext:Mention`. In this example it is linked both to local name and to a remote name. This implies that the names are the same (see Rule later).

Also, during the second step, the TNU is linked to the reified taxon concept label *Heser stoevi* sec. 10.3897/BDJ.4.e10095 via `:taxonConceptLabel` as even though the character content of the TNU does not contain a "sec.", we know for certain which concept the author is invoking as we are in the treatment title (current concept/ *this* concept).

```
<<Examples>>=

:heser-stoevi-nomenclature-heading po:contains :heser-stoevi-tnu .

:heser-stoevi-tnu a :TaxonomicNameUsage ;
  dc:date "2016-08-31"^^xsd:date ;
  cnt:chars
  "Heser stoevi urn:lsid:zoobank.org:act:E4D7D5A0-D649-4F5E-9360-D0488D73EEE8 Deltschev sp. n." ;
  dwc:genus "Heser" ;
  dwc:species "stoevi" ;
  dwc:scientificNameId "urn:lsid:zoobank.org:act:E4D7D5A0-D649-4F5E-9360-D0488D73EEE8 (ZooBank)" ;
  dwc:scientificNameAuthorship "Deltschev" ;
  dwc:taxonomicStatus "sp. n." ; dwciri:taxonomicStatus :TaxonDiscovery ;
  dwc:nameAccordingToId "10.3897/BDJ.4.e10095" ;

  :scientificName :heser-stoevi-deltshev ;
  :nameAccordingTo <http://dx.doi.org/10.3897/BDJ.4.e10095> ;
  :taxonConceptLabel :heser-stoevi-sec-deltshev ;

  :scientificName <http://zoobank.org/urn:lsid:zoobank.org:act:E4D7D5A0-D649-4F5E-9360-D0488D73EEE8> .

:heser-stoevi-deltshev owl:sameAs <http://zoobank.org/urn:lsid:zoobank.org:act:E4D7D5A0-D649-4F5E-9360-D0488D73EEE8>
@
```

Biological Taxonomy and Systematics

In this subsection we introduce classes and properties which are used to convey information from the domain of biological systematics.

Biological Names

In OpenBiodiv, we reify biological names.

In our conceptualization, taxa in nature are things (referents) that are refered to by our thoughts, theories and concepts (references) that are labeled or symbolized by by biological names (semiotic triangle).

Biological names play a dual role, however, in our system as they are also concepts, i.e. references of taxonomic name usages. A biological name may symbolize more than one taxon concept. It is useful to think of biological names then as taxon concept lineages. More about taxon concepts later.

Biological names have been modeled elsewhere such as for example in NOMEN. However, NOMEN takes the approach of using non-human-readable identifiers and only relying on labels to identify classes of taxonomic names, which does not fit our workflow. For example, the identifier for the class "biological name" is `NOMEN_0000030` . In our workflow both RDF generation and debugging would be severely hampered by this convention. That's why we have defined names in OpenBiodiv and mapped them to their NOMEN equivalents.

Def. (Biological Name, Scientific Name, Vernacular Name): *Biological Name, Scientific Name, and Vernacular Name are introduced as their NOMEN equivalents.*

```
<<Model of Biological Systematics>>=

:BiologicalName rdf:type owl:Class ;
    rdfs:label "Biological Name"@en ;
    owl:sameAs nomen:NOMEN_0000030 .

:ScientificName rdf:type owl:Class ;
```

```
    rdfs:subClassOf :BiologicalName ;
    rdfs:label "Scientific Name"@en ;
    owl:sameAs  nomen:NOMEN_0000036 .

:VernacularName a owl:Class ;
  rdfs:subClassOf :BiologicalName ;
  rdfs:label "Vernacular Name"@en ;
  owl:sameAs nomen:NOMEN_0000037 .
@
```

Def. (Taxon Concept Label): *We further introduce the class of taxon concept labels, unknown to NOMEN that is a biological name plus a reference to its description, i.e. it is the label of a taxon concept. A taxon concept label is a taxonomic name usage accompanied by an additional part, consisting of "sec." + an identifier or a literature reference of a work containing the expression of a taxon concept (for example a treatment).*

```
<<Model of Biological Systematics>>=

:TaxonConceptLabel rdf:type owl:Class ;
  rdfs:subClassOf :BiologicalName ;
  rdfs:label "Taxon Concept Label"@en ;
  rdfs:comment "A taxon concept label is a taxonomic name
usage accompanied by an additional part, consisting of 'sec.' + an identifier
or a literature reference of a work containing the expression of a taxon concept
(treatment)."@en .
@
```

We do not model scientific names down to the level of the Codes as NOMEN does. For example we do not make a distinction between a zoological and a botanical name. Nothing prevents us, however, from creating derived classes later on. This means that our model is somewhat crude but compatible with NOMEN.

For properties of biological names we take a different path from NOMEN. We also use different sets of properties to define relationships between biological names and for their data properties.

For data properties we use DwC terms.

To connect different biological objects such as taxon concepts or occurrences to a scientific name we use `:scientificName`, which is derived from `dwciri:scientificName`. Even though `dwciri:scientificName` is defined in spirit in http://rs.tdwg.org/dwc/terms/guides/rdf/index.htm#2.5_Terms_in_the_dwciri:_namespace,

we couldn't actually find a formal definition in RDF, that's why we're introducing it here together with a super-property to refer to a more broader class of names.

Def. (has biological name, has scientific name, has vernacular name):.

```
<<Model of Biological Systematics>>=

dwciri:scientificName rdf:type owl:ObjectProperty ;
  rdfs:label "scientific name"@en;
  rdfs:comment "the IRI version of dwc:scientificName"@en .

dwciri:nameAccordingTo rdf:type owl:ObjectProperty ;
  rdfs:label "name according to" ;
  rdfs:comment "the IRI version of dwc:scientificName"@en .

:biologicalName rdf:type owl:ObjectProperty ;
  rdfs:subClassOf pkm:mentions ;
        rdfs:label "mentions biological name"@en ;
        rdfs:range :BiologicalName .

:vernacularName rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf :biologicalName ;
        rdfs:label "mentions vernacular name" @en ;
        rdfs:range :VernacularName .
```

```
:scientificName rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf dwciri:scientificName, :biologicalName ;
  rdfs:label "mentions scientific name"@en ;
  rdfs:range :ScientificName ;
  rdfs:comment "'the scientific name property, derived from ':biologicalName', 'pkm:mentions', and 'dwciri:scientific

:taxonConceptLabel rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf :biologicalName ;
  rdfs:label "mentions taxon concept label"@en ;
  rdfs:range :TaxonConceptLabel .

:nameAccordingTo rdf:type owl:ObjectProperty ;
  rdfs:label "sec."@en ;
  rdfs:range frbr:Expression ;
  rdfs:comment "The reference to the source in which the specific taxon concept circumscription is defined or implied
@
```

For relationships between names we introduce two types of relationships: unidirectional and bidirectional.

Def. ('has related name'): *'has related name' is an object property that we use in order to indicate that two biological names are related somehow. This relationship is purposely vague as to encompass all situations where two biological names co-occur in a text. It is transitive and reflexive.*

```
<<Model of Biological Systematics>>=

:relatedName rdf:type owl:ObjectProperty, owl:TransitiveProperty, owl:ReflexiveProperty ;
  rdfs:label "has related name"@en ;
  rdfs:domain :BiologicalName ;
  rdfs:range :BiologicalName ;
  rdfs:comment "'has related name' is an object property that we
use in order to indicate that two biological names are related somehow. This
relationship is purposely vague as to encompass all situations where two
biological names co-occur in a text. It is transitive and reflexive."@en.

@
```

Def. (has replacement name): *This is a uni-directional property. Its meaning is that one one biological name links to a different biological name via the usage of this property, then the object of the triple is the form of the biological name the use of which is more accurate and should be preferred given the information that system currently holds. This property is only defined for scientific names.*

```
<<Model of Biological Systematics>>=

:replacementName rdf:type owl:ObjectProperty ,
                          owl:TransitiveProperty ;
  rdfs:label "has replacement name"@en ;
  rdfs:domain :ScientificName ;
  rdfs:range :ScientificName ;
  rdfs:comment "This is a uni-directional property. Its meaning
is that one one biological name links to a different biological name via the
usage of this property, then the object of the triple is the form of the
biological name the use of which is more accurate and should be preferred
given the information that system currently holds. This property is only
defined for scientific names."@en.
@
```

Now we define some rules for names

Rule 1 for Names: *For a scientific name X, if there doesn't exist a TNU mentioning X, which has the taxon status of `:UnavailableName`, or if there does exist a TNU Y mentioning X with the status of `:UnavailableName`, but there also exists a TNU Z mentioning X with a later date than Y, which has the status of `:AvailableName` or `:ReplacementName`, then X has the taxon status of `:AvailableName`.*

```
<<Rules>>=

# rules need to be evaluated in the order here
# I. set all names that have not been made unavailabel to available
INSERT {
  ?a dwciri:taxonomicStatus :AvailableName .
}
WHERE {
  ?a rdf:type :ScientificName .
  ?tnu pkm:mentions ?a .
  UNSAID { ?tnu dwciri:taxonomicStatus :UnavailableName .}
}

 # note the date here refers to when was the taxonomic status of the name last determined
 # 2. set all names that were made unavailable at one point to unavailable and copy the
 # date
INSERT {
  ?a dwciri:taxonomicStatus :UnavailableName .
  ?a dc:date ?d .
}
WHERE {
  ?a rdf:type :ScientificName .
  ?tnu pkm:mentions ?a ;
       dwciri:taxonomicStatus :UnavailableName ;
       dc:date ?d .
}

 # 3. set names to :Available back
DELETE {
  ?a dwciri:taxonomicStatus :UnavailableName ;
  ?a dc:date ?d0 .
}
INSERT {
  ?a dwciri:taxonomicStatus :AvailableName .
  ?a dc:date ?d1 .
}
WHERE {
  ?a rdf:type :ScientificName ;
     dc:date ?d0 .
  ?tnu1 pkm:mentions ?a ;
        dwciri:taxonomicStatus :AvailableName ;
        dc:date ?d1 .
  FILTER ( ?d1 > ?d0 )
}

DELETE {
  ?a dwciri:taxonomicStatus :UnavailableName ;
  ?a dc:date ?d0 .
}
INSERT {
  ?a dwciri:taxonomicStatus :AvailableName .
  ?a dc:date ?d1 .
}
WHERE {
  ?a rdf:type :ScientificName ;
     dc:date ?d0 .
  ?tnu1 rdf:type :TaxonomicNameUsage ;
        pkm:mentions ?a ;
        dwciri:taxonomicStatus :ReplacementName ;
        dc:date ?d1 .
  FILTER ( ?d1 > ?d0 )
}
@
```

Rule 2 for Names: *For a scientific name X, if it is mentioned in the heading of a nomenclature section (treatment title) in a TNU Y with status* `:ReplacementName` *, then every name Z_i, mentioned in the nomenclatural citation list in TNU's with status* `:UnavailableName` *is linked to X via* `:replacementName` *.*

```
<<Rules>>=

# II. Link replacement names
INSERT {
    ?a trt:replacementName ?b .
}
WHERE {
        ?a a :ScientificName .
        ?b a :ScientificName .
        [] a trt:Nomenclature ;
                   po:contains [ a :TaxonomicNameUsage ;
                                              pkm:mentions ?a ;
                                              dwciri:taxonomicStatus :ReplacementName ] ,
                                    [ a :TaxonomicNameUsage ;
                                  pkm:mentions ?b ;
                                  dwciri:taxonomicStatus :UnavailableName ].
}
@
```

Rule 3 for Names: *All names in the nomenclature section are linked via* `:relatedName` .

```
<<Rules>>=

# III. if two names are mentioned in the same nomenclature
# section then they are related

INSERT {
    ?a trt:relatedName ?b .
}
WHERE {
        ?a a :ScientificName .
        ?b a :ScientificName .
        ?c a trt:Nomenclature ;
           po:contains [ pkm:mentions ?a ];
                          [ pkm:mentions ?b ].
}
@
```

Rule 4 for Names: *If for a name X, there exists a TNU Y If a TNU is marked as* `:Conserved` *, then the name is also marked as* `:Conserved` *. A conserved name should not be made* `:Unavailable` *!*

```
<<Rules>>=

# IV. Conserved names

INSERT {
    ?a dwciri:taxonomicStatus :ConservedName .
}
WHERE {
  ?a a :ScientificName .
  ?t a :TaxonomicNameUsage ;
     pkm:mentions ?a
     dwciri:taxonomicStatus :ConservedName .
@
```

Rule 5 for Names: *If a TNU points to two different names with* `dwciri:scientificName` *, then they are the same:*

```
<<Rules>>=

# V. Conserved names

INSERT {
    ?name1 owl:sameAs ?name2 .
}
```

```
WHERE {
  ?name1 a :ScientificName .
  ?name2 a :ScientificName .
  ?t a :TaxonomicNameUsage ;
   t dwciri:scientificName ?name1, ?name2 .
  FILTER( ?name1 != ?name2 ) .
@
```

Example We go back to the example of *Heser stoevi*. The meaning of the date property here is to indicate when was the taxonomic status assumed.

```
<<Examples>>=

:heser-stoevi-deltshev a :ScientificName ;
        skos:prefLabel "Heser stoevi Deltshev" ;
    dwc:species "stoevi" ;
    dwc:genus "Heser" ;
    dwc:taxonRank "species" ;
    dwciri:taxonomicStatus <http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/accepted> ;
    dwc:scientificNameAuthorship "Deltschev" ;
    dc:date "2016-08-31"^^xsd:date .
@
```

Example. Let's take another example, the paper http://bdj.pensoft.net/articles.php?id=8030&instance_id=2809105. From it, we can say:

```
<<Examples>>=

:nomenclature-bdje8030 a trt:Nomenclature ;
  po:contains :nomenclature-heading-bdje8030, :cit-list-bdje8030 .

:nomenclature-heading-bdje8030 a :NomenclatureHeading ;
  po:contains :harmonia-manillana-tnu-heading .

:cit-list-bdje8030 a trt:NomenclatureCitationList ;
  po:contains :leis-papuensis-tnu-citation .

:harmonia-manillana-tnu-heading a :TaxonomicNameUsage ;
  dc:date "2016-08-16"^^xsd:date ;
  cnt:chars "Harmonia manillana (Mulsant, 1866)" .

:leis-papuensis-tnu-citation a :TaxonomicNameUsage ;
  dc:date "2016-08-16"^^xsd:date ;
  cnt:chars "Leis papuensis var. suffusa Crotch 1874 121 (Lectotype, UCCC). Korschefsky 1932 : 275.- Gordon 1987 : 14
  dwc:taxonomicStatus "var. suffusa Crotch 1874 121 (Lectotype, UCCC). Korschefsky 1932 : 275.- Gordon 1987 : 14 (lec
  dwciri:taxonomicStatus :Unavailable .

:harmonia-manillana-mulsant-1866 a :ScientificName ;
  skos:prefLabel "Harmonia manillana (Mulsant, 1866)" ;
  skos:altLabel "Harmonia manillana" ;
  dwc:species "manillana" ;
  dwc:genus "Harmonia" ;
  dwc:taxonRank "species" ;
  dwc:scientificNameAuthorship "(Mulsant, 1866)" ;
  dc:date "2016-08-16"^^xsd:date ;
  dwc:taxonomicStatus :Available ;
  :relatedName :leis-papuensis .

:leis-papuensis a :ScientificName ;
  skos:prefLabel "Leis papuensis" ;
  dwc:species "papuensis" ;
  dwc:genus "Leis" ;
  dwc:taxonRank "species" ;
  dc:date "2016-08-16"^^xsd:date ;
  dwciri:taxonomicStatus :Unavailable ;
  :replacementName :harmonia-manillana-mulsant-1866 ;
```

```
  :relatedName :harmonia-manillana-mulsant-1866 .
@
```

Taxon Concepts

Discussion. Our view of taxon concepts is based on Berendsohn (1995) and Franz et al (2008).

We consider any given taxon concept to be a scientific theory (concept) about a class of biological organisms (taxon). The class description, as in "as in all spiders have spinnerets (silk-producing glands)" (Nico Franz, personal correspondence), is called *intensional meaning,* whereas the group of organisms in nature conforming with the intensional meaning is called the class *extension*.

We want to model both the intensional meaning (which traits do organisms belonging to a taxon have) and the extension of taxon concepts (which organisms belong to a taxon) and the extensions being organisms that are considered to be members of the class.

This necessitates the view that taxon concepts are both instances of a taxon concept class and are classes of ogranisms. Later, we will show that this means that we model Taxon Concepts with OWL Full.

OpenBiodiv taxon concepts are instances of `dwc:Taxon` and vice versa (*"A group of organisms [sic] considered by taxonomists to form a homogeneous unit."*).

Also, taxon concepts are instances of `frbr:Work` as well, but not vice versa (*"A distinct intellectual or artistic creation. A work is an abstract entity; there is no single material object one can point to as the work. We recognize the work through individual realizations or expressions of the work, but the work itself exists only in the commonality of content between and among the various expressions of the work. When we speak of Homer's Iliad as a work, our point of reference is not a particular recitation or text of the work, but the intellectual creation that lies behind all the various expressions of the work."*).

Furthermore, taxon concepts can also be modeled as `skos:Concept` , but not vice versa (*"A SKOS concept can be viewed as an idea or notion; a unit of thought. However, what constitutes a unit of thought is subjective, and this definition is meant to be suggestive, rather than restrictive."*).

All three classes represent a distinctive view that we want to adopt in modeling different features of taxon concepts.

Holding the views of Berendsohn and of Franz, we require that each taxon concept is linked to both a biological name and to a work (i.e. publication, database, etc.), where the circumscription is properly defined.

Def. (Taxon Concept):

```
<<Model of Biological Systematics>>=

:TaxonConcept rdf:type owl:Class ;
  owl:equivalentClass dwc:Taxon ;
  rdfs:subClassof frbr:Work , skos:Concept ,
                  [ rdf:type owl:Restriction ;
                    owl:onProperty :taxonConceptLabel ;
                    owl:minCardinality "1" ] .
@
```

Example. In the next example we introduce the concept of *Heser stoevi* according to the article published by Deltshev in 2016. First, we introduce an instance of `:TaxonConcept` and link this instance to the scientific name *Heser stoevi* via the appropriate DwC term. Next, we establish a link between the significant bibliographic unit (in this case journal article) containing the treatment, which is the realization of the taxon concept. The last point we would like to make is that the taxon concept label, which is in this case `Heser stoevi sec. 10.3897/BDJ.4.e100095` is constructed by pasting together the label of the biological name and the expression that are assigned to the concept glued together by `sec.` .

```
<<Examples>>=

:heser-stoevi-sec-deltshev a :TaxonConceptLabel ;
   skos:prefLabel "Heser stoevi Deltshev sec. 10.3897/BDJ.4.e10095" ;
   dwc:species "stoevi" ;
   dwc:genus "Heser" ;
   dwc:taxonRank "species" ;
   dwc:scientificNameAuthorship "Deltschev" ;
   ::nameAccordingTo <http://dx.doi.org/10.3897/BDJ.4.e10095> .

:concept-deltshev-2016 a :TaxonConcept ;
  :taxonConceptLabel :heser-stoevi-sec-deltshev .

:heser-stoevi-sec-gbif20170323 a :TaxonConceptLabel ;
  skos:prefLabel "Heser stoevi sec. doi:10.15468/39omei" ;
  dwciri:scientificName :heser-stoevi-deltshev ;
  dwciri:nameAccordingTo :gbif20170323 .

:concept-gbif a :TaxonConcept ;
  :taxonConceptLabel :heser-stoevi-sec-gbif20170323 .

<http://dx.doi.org/doi:10.15468/39omei> a fabio:Database ;
  skos:prefLabel "GBIF Backbone Taxonomy" ;
  skos:altLabel "doi:10.15468/39omei" ;
  prism:doi "doi:10.15468/39omei" ;
  dc:date "2017-03-23"^^xsd:date ;
  rdfs:comment "A dump of GBIF's backbone taxonomy on 23 Mar 2017."@en ;
  po:contains [ a :TaxonConceptLabel ;
                dc:date "2017-03-23"^^xsd:date ;
                dwc:scientificName "Heser stoevi Deltshev, 2016" ;
                dwc:nameAccordingTo "GBIF Backbone Taxonomy" ;
                dwciri:nameAccordingTo <http://dx.doi.org/doi:10.15468/39omei> ;
                dwciri:scientificName :heser-stoevi-deltshev ;
                pkm:mentions :heser-stoevi-sec-gbif20170323  ] .
@
```

Note that in the above example one scientific name, *Heser stoevi*, is linked to two different taxon concept labels, as one taxon concept label denotes the concept coming from the article and the other one comes from the GBIF database.

It is possible to express that these are the same thing, that one is a subconcept of the other, or even more granular relationships.

Taxon Concept Relationships

Example congruence 1. If we want to express that two concepts are exactly the same both intensionally and ostensively then we use `owl:sameAs` . However, if we want to express that the taxon concepts have the same extension without being equal intensionally (as in the spider example), then we use `owl:equivalentClass` .

```
<<Examples>>=

:heser-stoevi-sec-deltschev owl:sameAs :heser-stoevi-sec-gbif20170323 .
@
```

Note that this will copy both taxon concept labels both ways, but this is OK as the different taxon concept labels refer to the same class.

Example congruence 2. Let's define the two spider concepts here

```
<<Examples>>=

:haveSpinnerets rdf:type owl:DatatypeProperty ;
  rdfs:domain :TaxonConcept ;
  rdfs:comment "This property if true, indicates that some taxon has spinnertes." .
```

```
:havePedipals rdf:type owl:DataTypeProperty ;
  rdfs:domain :TaxonConcept ;
  rdfs:comment "If this property is true, that the taxon has pedipals" .

:spiders-with-spinnerets-sec-rdfguide a :TaxonConceptLabel ;
  dwc:order "Araneae" ;
  skos:prefLabel "Araneae sec. OpenBiodiv RDF Guide" ;
  :nameAccordingTo <https://github.com/pensoft/OpenBiodiv/blob/master/Ontology/RDF_Guide.md> .

:spiders-with-pedipals-sec-rdfguide a :TaxonConceptLabel ;
  dwc:order "Aranea" ;
  skos:prefLabel "Araneae sec. OpenBiodiv RDF Guide" ;
  :nameAccordingTo <https://github.com/pensoft/OpenBiodiv/blob/master/Ontology/RDF_Guide.md> .

:concept-spinnerets a :TaxonConcept ;
  :taxonConceptLabel :spiders-with-spinnerets-sec-rdfguide ;
  :haveSpinnerrets "true"^^xsd:boolean .

:concept-pedipals a :TaxonConcept ;
  :taxonConceptLabel :spiders-with-pedipals-sec-rdfguide ;
  :havePedipals "true"^^xsd:boolean .

:spider1 a :concept-spinnerts .
:spider2 a :concept-pedipals .

:concept-spinnerts owl:equivalentClass :concept-pedipals .
@
```

Here the implication is that although the intensional meaning of the two concepts is different, they have the same class extension. I.e. we will infer

```
:spider1 a :concept-pedipals;
:spider2 a :concept-spinnerts'
```

but we will not copy any of the `:havePedipals` or `:haveSpinnerts` to the other concept.

Example of contained concepts. For contained concept we use `rdfs:subClassOf` :

```
<<Examples>>=

:animalia-sec-gbif a :TaxonConceptLabel ;
  skos:prefLabel "Animalia sec. GBIF Backbone Taxonomy" ;
  dwciri:scientificName :animalia ;
  :nameAccordingTo <http://dx.doi.org/doi:10.15468/39omei> .

:animalia a :ScientificName ;
  dwc:scientificName "Animalia" ;
  dwc:taxonRank "kingdom" ;
  dc:date "2017-03-23"^^xsd:date ;
  dwciri:taxonomicStatus :Available .

:concept-animalia-gbif a :TaxonConcept ;
  dwciri:scientificName :animalia ;
  dwc:taxonId "1 (GBIF)" .

:heser-stoevi-sec-gbif20170323 rdfs:subClassOf :animlia-sec-gbif20170323 .
@
```

Example (Relatedness) If two taxon concepts are related we can use `skos:related` .

```
<<Examples>>=

:heser-nicola a :ScientificName ;
  dwc:scientificName "Heser nilicola (O. P.-Cambridge, 1874)" ;
```

```
  skos:prefLabel "Heser nilicola" ;
  dwc:genus "Heser" ;
  dwc:species "nilicola" ;
  dwc:taxonRank "species" .

:heser-nicola-sec-unibe a :TaxonConceptLabel ;
  skos:prefLabel "Heser nilicola (O. P.-Cambridge, 1874) sec. Unibe" ;
  dwciri:scientificName :heser-nicola ;
  frbr:realization <http://www.araneae.unibe.ch/data/3301> .

:concept-heser-nicola-unibe a :TaxonConcept ;
  :taxonConceptLabel :heser-nicola-sec-unibe .

:heser-stoevi-deltschev-sec-deltschev skos:related :heser-nicola-sec-unibe .
@
```

Complex Relationships with RCC-5

**Def.: *A set of properties describing RCC-5 relations.*

```
<<Model of Biological Systematics>>=

:RCC5Statement rdf:type owl:Class ;
  rdfs:label "RCC5 Statement" ;
  rdfs:comment "A statemnt of RCC-5 relationship" .

:rcc5Property rdf:Type owl:ObjectProperty ;
  rdfs:domain :RCC5Statement .

:rcc5fromRegion rdf:type owl:ObjectProperty ;
  rdfs:subPropertyOf :rcc5Property ;
  rdfs:label "from region" ;
  rdfs:comment "Connects to the RCC5 statement to the originating region"@en .

:rcc5toRegion rdf:type owl:ObjectProperty ;
  rdfs:label "to region" ;
  rdfs:subPropertyOf :rcc5Property ;
  rdfs:comment "Connects to the RCC5 statement to the target region"@en .

:rcc5RelationType rdf:type owl:ObjectProperty ;
  rdfs:label "relation type" ;
  rdfs:subPropertyOf :rcc5Property ;
  rdfs:range :RCC5Relation ;
  rdfs:comment "Connects an RCC-5 statement to the type of RCC-5 relation between the regions."@en .
@
```

In order to model complex RCC-5 statements such as those made in Jansen & Franz 2015 we use the above defitions and the RCC5 Vocabulary described in the Appendix.

We model the following exceprt here:

"Minyomerus microps (Say, 1831: 9) sec. Jansen & Franz (2015), stat. n. == (INT) AND > (OST) AND = Minyomerus innocuus Horn, 1876: 18 sec. Horn (1876) (type, designated by Pierce 1913: 400), syn. n."

```
<<Examples>>=

:jansen-franz-2015 a fabio:JournalArticle ;
  skos:prefLabel "10.3897/zookeys.528.6001" .

:concept-minyomerus-microps-jansen-franz a :TaxonConcept .

:concept-minyomerus-innocuus-horn a :TaxonConcept .

:microps-innocuus-relation-int a :RCC5Stament ;
  :rcc5FromRegion :concept-minyomerus-microps-jansen-franz ;
  :rcc5ToRegion :concept-minyomerus-innocuus-horn ;
```

```
  :rcc5RelationType :Equals_INT ;
  frbr:expression :jansen-franz-2015 .

:microps-innocuus-relation-ost a :RCC5Stament ;
  :rcc5FromRegion :concept-minyomerus-microps-jansen-franz ;
  :rcc5ToRegion :concept-minyomerus-innocuus-horn ;
  :rcc5RelationType :InverseProperPart_OST ;
  frbr:expression :jansen-franz-2015 .
@
```

The rest of Biodiversity Sysmtematics

We follow the Darwin-SW model.

Apendicies

Vocabulary of Taxonomic Statuses

Taxonomic name usages (TNU's) in taxonomic articles may be accompanied by strings such as "new. comb.", "new syn.", "new record for Cuba", and many others. These postfixes to a taxonomic name usage are called in our model taxonomic statuses and have taxonomic as well nomenclatural meaning. For example, if we are describing a species new to science, we may write "n. sp." after the species name, e.g. "*Heser stoevi* Deltchev sp. n." This particular example is also a nomenclatural act in the sense of the Codes of zoological or botanical nomenclature.

Not all statuses are necessarily nomenclatural in nature. Sometimes the status is more of a note to the reader and conveys taxonomic rather than nomenclatural information. E.g. when a previously known species is recorded in a new location.

Here we take the road of modeling statuses from the bottom-up, i.e. based on their actual use in three of the most successful journals in biological systematics - ZooKeys, Biodiversity Data Journal, and PhytoKeys. We have analyzed about 4,000 articles from these journals and have come up with a vocabulary of statuses described below. The terms in this vocabulary refer to broad concepts and encompass both specific cases of botanical or zoological nomenclature as well as purely taxonomic or informative use. We believe these concepts to be adequately granular for the purposes of reasoning in OpenBiodiv (see the paragraphs on rules in biological names.

Extraction of 4,000 TNU suffixes

1. The first step we took in creating this vocabulary is to extract the actual real-world taxonomic statuses as found almost 4,000 Pensoft articles. The script to achieve this is found under mine_tnu.R. The output of the script is a list of taxonomic status abbreviations as a text file, taxonomic_statuses.txt, in total 253 distinct usages.
2. The second step we took is to normalize these statuses manually: clean up parts of abbreviations that are not part of the status (such as particular taxon names) or remove irrelevant statuses (such as geological information).
3. After cleaning up the usages we came up with the terms in the next section and assigned each of remaining status abbreviations to one of the terms. The assignments can be seen in text files with correspondning names.

For a similar attempt, see http://rs.gbif.org/vocabulary/gbif/taxonomic_status.xml. We do map our statuses to the GBIF statuses to the best of our knowledge but we have more statuses.

Formal Vocabulary

```
<<Vocabulary of Taxonomic Statuses>>=

:TaxonomicStatus rdf:type owl:Class ;
  rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty <http://www.w3.org/2004/02/skos/core#inScheme> ;
                    owl:someValuesFrom :TaxonomicStatusTerms ] ;
```

```
  rdfs:label "Taxonomic Status"@en ;
  rdfs:comment "The status following a taxonomic name usage in a taxonomic
               manuscript, i.e. 'n. sp.',
                                'comb. new',
                                'sec. Franz (2017)', etc"@en .

:TaxonomicStatusTerms rdf:type owl:Class ;
  rdfs:subClassOf <http://www.w3.org/2004/02/skos/core#ConceptScheme> ,
                               [ rdf:type owl:Restriction ;
                                 owl:onProperty fabio:isSchemeOf ;
                                 owl:allValuesFrom :TaxonomicStatus] ;
  rdfs:label "OpenBiodiv Vocabulary of Taxonomic Statuses"@en ;
  fabio:hasDiscipline <http://dbpedia.org/page/Taxonomy_(biology)> .

  <<Taxonomic Uncertainty>>
  <<Taxon Discovery>>
  <<Replacement Name>>
  <<Unavailable Name>>
  <<Available Name>>
  <<Type Specimen Designation>>
  <<Type Species Designation>>
  <<New Occurrence Record>>
  <<Taxon Concept Label>>

@
```

Taxonomic Uncertainty

When a TNU is followed by the term `:TaxonomicUncertainty`, the implication is that the taxon concept identified by the name has an uncertain placement in the biological taxonomy, or if we are talking about a specimen or a sample of some kind that we are unable to identify the taxon down to its rank. This is related to http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/accepted

```
<<Taxonomic Uncertainty>>=

:TaxonomicUncertainty a :TaxonomicStatus ;
  rdfs:label "Taxonomic Uncertainty"@en ;
  skos:inScheme :TaxonomicStatusTerms ;
  skos:related <http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/accepted> ;

  rdfs:comment "The implication of this term that the taxon concept identified
by a name has an uncertain placement in the biological taxonomy, or if we
are talking about a specimen or a sample of some kind that we are unable to
identify the taxon down to its rank. "@en .

@
```

Here're some ways in which it can be abbreivated:

taxonomic_uncertainty.txt

Taxon Discovery

When a TNU is followed by the term `:TaxonDiscovery`, the implication is that the present context in which the TNU is used is circumscribing a new taxon concept (a taxon concept of a taxon new to science), and simultaneously assigning it a new name.

```
<<Taxon Discovery>>=

:TaxonDiscovery a :TaxonomicStatus ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:label "Taxon Discovery"@en ;

  rdfs:comment "When a TNU is followed by the term `:TaxonDiscovery`, the
implication is that the present context in which the TNU is used is
```

```
circumscribing a new taxon concept (a taxon concept of a taxon new to
science) and simultaneously assigning it a new name."@en .

@
```

Here're some ways in which it can be abbreivated:

taxon_discovery.txt

Replacement Name

When a TNU is followed by the term `:ReplacementName`, the implication is that the name that is referred to by the TNU is replacing another name for various reasons, and thus becoming the preferred/accepted/available way of refering to whatever taxon concepts the now replaced name had been referring to. Cases include changes of rank, new combinations, spelling mistakes, etc.

```
<<Replacement Name>>=

:ReplacementName a :TaxonomicStatus ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:label "Replacement Name"@en ;
  rdfs:comment "When a TNU is followed by the term `:ReplacementName`, the implication is that
the name that is referred to by the TNU is replacing another name for various
reasons, and thus becoming the preferred/accepted/available way of refering to
whatever taxon concepts the now replaced name had been referring to. Cases
include changes of rank, new combinations, spelling mistakes, etc.
"@en .

@
```

Here're some ways in which it can be abbreivated:

replacement_name.txt

Unavailable Name

When a TNU is followed by the term `:UnavailableName`, the implication is that the name that is being referred to by the TNU is no longer or has never been available for use due to the fact that it has either been replaced or it has been determined that the name has been improperly coined or published, or the name contains any general error rendering it unfit for use. This is the same as http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/synonym

```
<<Unavailable Name>>=

:UnavailabeName a :TaxonomicStatus ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:label "UnavailableName"@en ;
  skos:exactMatch  <http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/synonym> ;
  rdfs:comment "When a TNU is followed by the term `:UnavailableName`, the implication is that
the name that is being referred to by the TNU is no longer or has never been
available for use due to the fact that it has either been replaced or it has
been determined that the name has been improperly coined or published, or the
name contains any general error rendering it unfit for use."@en .
@
```

Here're some ways in which it can be abbreivated:

unavaible_name.txt

Available Name

When a TNU is followed by the term `:AvailableName`, the implication is that the name that is being referred to by the TNU has been determined to be fit for use either by revoking an older act rendering the name unavailable or by fixing

other issues with the name or finding out that other issues with the name had been fixed, or just stating the fact that the name shall be used or even conserving it, so that the name can be freely used from then on in compliance with all Codes and practices. This is the same as http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/accepted

```
<<Available Name>>=

:AvailableName a :TaxonomicStatus ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:Label "Available Name"@en ;
  skos:exactMatch <http://rs.gbif.org/vocabulary/gbif/taxonomicStatus/accepted> ;
  rdfs:comment "When a TNU is followed by the term `:AvailableName`, the
implication is that the name that is being referred to by the TNU has been
determined to be fit for use either by revoking an older act rendering the
name unavailable or by fixing other issues with the name or finding out that
other issues with the name had been fixed, or just stating the fact that the
name shall be used or even conserving it, so that the name can be freely used
from then on in compliance with all Codes and practices."@en.
@
```

available_name.txt

Type Species Designation

When a TNU is followed by the term `:TypeSpeciesDesignation`, the implication is that the taxon concept of the name in the TNU as understood by the author should be considered the type species of a higher taxon as understood by the author.

```
<<Type Species Designation>>=

:TypeSpeciesDesignation a :TaxonomicStatus ;
  rdfs:label "Type Species Designation" @en ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:comment "When a TNU is followed by the term `:TypeSpeciesDesignation`, the implication
is that the taxon concept of the name in the TNU as understood by the author
should be considered the type species of a higher taxon as understood by the
author."@en.

@
```

type_species_designation.txt

Type Specimen Designation

When a TNU is followed by the term `:TypeSpecimenDesigna`, the implication is that the specimen identified by the name in TNU should be considered a type of the taxon concept identified by the name as understood by the author of TNU.

```
<<Type Specimen Designation>>=

:TypeSpecimenDesignation a :TaxonomicStatus ;
  rdfs:label "Type Specimen Designation" @en ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:comment "When a TNU is followed by the term `:TypeSpecimenDesignation`, the implication
is that the specimen identified by the name in TNU should be considered a type
of the taxon concept identified by the name as understood by the author of
TNU."@en.

@
```

type_specimen_designation.txt

New Record

When a TNU is followed by the term `:NewRecord`, the implication is that the description of taxon concept of the name as understood by the author is being extended with new occurrences (for a given locality).

```
<<New Occurrence Record>>=

:NewOccurrenceRecord a :TaxonomicStatus ;
  rdfs:label "New Occurrence Record" @en ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:comment "When a TNU is followed by the term `:NewRecord`, the implication is that the
description of taxon concept of the name as understood by the author is being
extended with new occurrences (for a given locality).
"@en.

@
```

new_occurrence_record.txt

Taxon Concept Label

Sometimes, incorrectly a taxon concept label (sec. Author (year) may be misunderstood and marked up) as a taxonomic status. This term has been created to indicate that a particular TNU is taxonomic concept label.

```
<<Taxon Concept Label>>=

:TaxonConceptLabel a :TaxonomicStatus ;
  skos:inScheme :TaxonomicStatusTerms ;
  rdfs:label "Taxon Concept Label"@en ;
  rdfs:comment "Sometimes, incorrectly a taxon concept label (sec. Author (year) may be
misunderstood and marked up) as a taxonomic status. This term has been created
to indicate that a particular TNU is taxonomic concept label.
"@en .

@
```

taxon_concept_label.txt

Vocabulary of RCC5 Terms

```
<<Vocabulary of RCC5 Terms>>=

:RCC5Relation rdf:type owl:Class ;
  rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty <http://www.w3.org/2004/02/skos/core#inScheme> ;
                    owl:someValuesFrom :RCC5RelationTerms ] ;
  rdfs:label "RCC5 Relation"@en ;
  rdfs:comment "The of RCC 5 relation, e.g. 'partially overlaps'"@en .

:RCC5RelationTerms rdf:type owl:Class ;
  rdfs:subClassOf <http://www.w3.org/2004/02/skos/core#ConceptScheme> ,
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty fabio:isSchemeOf ;
                                  owl:allValuesFrom :RCC5Relation] ;
  rdfs:label "OpenBiodiv Vocabulary of RCC5 Relations"@en .

:Equals_INT rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Equals (INT)" ;
  rdfs:comment "= EQ(x,y) Equals (intensional)"@en .

:ProperPart_INT rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Proper Part (INT)" ;
  rdfs:comment "< PP(x,y) Proper Part of (intensional)"@en .

:InverseProperPart_INT rdf:type :RCC5Relation ;
```

```
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Inverse Proper Part (INT)" ;
  rdfs:comment "iPP(x, y) Inverse Proper Part (intensional)"@en .

:PartiallyOverlaps_INT rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Partially Overlaps (INT)" ;
  rdfs:comment "o PO(x,y) Partially Overlaps (intensional)"@en .

:Disjoint_INT rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Disjoint (INT)" ;
  rdfs:comment "! DR(x,y) Disjoint from (intensional)."@en .

:Equals_OST rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Equals (OST)" ;
  rdfs:comment "= EQ(x,y) Equals (ostensive)"@en .

:ProperPart_OST rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Proper Part (OST)" ;
  rdfs:comment "< PP(x,y) Proper Part of (ostensive)"@en .

:InverseProperPart_OST rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Inverse Proper Part (OST)" ;
  rdfs:comment "iPP(x, y) Inverse Proper Part (ostensive)"@en .

:PartiallyOverlaps_OST rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Partially Overlaps (OST)" ;
  rdfs:comment "o PO(x,y) Partially Overlaps (ostensive)"@en .

:Disjoint_OST rdf:type :RCC5Relation ;
  skos:inScheme :RCC5RelationTerms ;
  rdfs:label "Disjoint (OST)" ;
  rdfs:comment "! DR(x,y) Disjoint from (ostensive)."@en .
@
```

Vocabulary of Paper Types

TODO: Future work

Def.: Pensoft's journals have some paper types, which we define herein. First of all, we introduce Paper Types as a Term Dictionary in the discipline of Bibliography. Then we introduce the different paper types as Subject Term's in the scheme of Paper Types. See the SPAR ontologies for more info on this this.

```
<<Vocabulary Paper Types>>=


:PaperTypeTerms rdf:type owl:Class ;
  rdfs:subClassOf <http://www.w3.org/2004/02/skos/core#ConceptScheme> ,
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty fabio:isSchemeOf ;
                                  owl:allValuesFrom :PaperType] ;
  rdfs:label "Paper Types Vocabulary"@en ;
  rdfs:comment "A list of paper (article) types published in Pensoft's
                journals"@en ;
  fabio:hasDiscipline dbpedia:Bibliography .

:PaperType rdf:type owl:Class ;
  rdfs:subClassOf [ rdf:type owl:Restriction ;
                    owl:onProperty <http://www.w3.org/2004/02/skos/core#inScheme> ;
                    owl:someValuesFrom :PaperTypeTerms ] ;
  rdfs:label "Paper Type"@en ;
  rdfs:comment "A Specific Type Of Paper"@en .
```

```
:SingleTaxonTreatment a PaperType ;
  rdfs:label "Single Taxon Treatment"@en;
  rdfs:comment "A type of paper with only one taxonomic treatment"@en ;
  skos:inScheme pensoft:PaperTypes .
@
```

TODO: Extract paper types.

Vocabulary of Taxon Classification

TODO: Future work

Def. 10 of controlled vocabulary (Taxon Classification): Pensoft, in its Keywords uses certain taxon names for the classification of its papers. These taxon names are borrowed from GBIF. Here we define a term dictionary analogously to paper types:

```
<<Vocabulary of Taxon Classification>>=

:TaxonClassificationTerms rdf:type owl:Class ;
  rdfs:subClassOf <http://www.w3.org/2004/02/skos/core#ConceptScheme> ,
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty fabio:isSchemeOf ;
                                  owl:allValuesFrom :TaxonClassification] ;
  rdfs:label "Taxonomic Classification"@en ;
  rdfs:comment "A list of taxon names borrowed for GBIF for the
                classification of papers."@en ;
  fabio:hasDiscipline dbpedia:Taxonomy .
@
```

Vocabulary of Chronological Classification

TODO: Future work

```
<<Vocabulary of Chronological Classification>>=
:ChronologicalClassificationTerms rdf:type owl:Class ;
  rdfs:subClassOf <http://www.w3.org/2004/02/skos/core#ConceptScheme> ,
                                [ rdf:type owl:Restriction ;
                                  owl:onProperty fabio:isSchemeOf ;
                                  owl:allValuesFrom :ChronologicalClassigication] ;
  rdfs:label "Chronological Classification"@en ;
  rdfs:comment "A vocabulary of chronological eras that can be used in
                Pensoft's journals"@en ;
  fabio:hasDiscipline dbpedia:Paleontology .
@
```

Directly imported external ontologies

Some third party ontologies cannot be imported via `owl:imports` for various reasons (updated versions that we don't use, broken links, etc.) We add the objects that we borrow from them here.

Parts of PROTON

```
<<Borrowed Parts from External Ontology>>=

ptop:Entity rdf:type owl:Class ;
            rdfs:comment "Any sort of an entity of interest, usually something existing, happening, or purely abstrac
            rdfs:label "Entity"@en .

ptop:Object rdf:type owl:Class ;
            rdfs:subClassOf ptop:Entity ;
```

```
            rdfs:comment "Objects are entities that could be claimed to exist - in some sense of existence. An object
            rdfs:label "Object"@en .

ptop:Statement rdf:type owl:Class ;
               rdfs:subClassOf ptop:Object ;
               rdfs:comment "A message that is stated or declared; a communication (oral or written), setting forth p
               rdfs:label "Statement"@en .

ptop:InformationResource rdf:type owl:Class ;
                         rdfs:subClassOf ptop:Statement ;
                         rdfs:comment "InformationResource denotes an information resource with identity, as defined
                         rdfs:label "Information Resource"@en .

pkm:mentions
      rdf:type owl:ObjectProperty ;
      rdfs:comment "A direct link between an information resource, like a document or a section and an entity." ;
      rdfs:domain ptop:InformationResource ;
      rdfs:label "mentions" ;
      rdfs:range ptop:Entity .

pext:Mention rdf:type owl:Class ;
             rdfs:subClassOf ptop:InformationResource ;
             rdfs:comment "An area of a document that can be considered a mention of something."@en ;
             rdfs:label "Section" .
@
```