

Visual Terminology Alignment Tool Specifications

Version 3.0

July 2018

Anastasia Axaridou, R & D, FORTH-ICS

Contents

1	Introduction.....	3
2	The alignment problem in VisTA	3
3	General features of the system	4
3.1	Basic operations	4
3.2	Alignment process	4
3.3	Visualisation of the terminologies.....	4
3.3.1	Preferred languages	4
3.4	Widget template configuration	5
4	User interface and functions	6
4.1	Import/Export: managing datasets	6
4.2	Edit: loading/creating/updating alignments	7
4.2.1	Loading terminology trees.....	7
4.2.2	Searchable trees	8
4.2.3	Representation of the aligned terms	8
4.2.4	Collapsible, selectable, excludable terms.....	9
4.2.5	Draggable terms	9
4.2.6	Viewing the alignment correspondences	9
4.2.7	Highlighted terms – Notation	9
4.2.8	Deleting alignment correspondences.....	10
4.2.9	Refreshing source hierarchy and the alignment graph	10
4.2.10	Information about a term.....	10
4.3	Searching for terms	10
4.4	Alignment rules	11
4.5	Multi-user operation	11
5	Alignment graph	12
	Annex 1 – The main widget template (Version 5)	13

1 Introduction

The alignment of terminologies can be considered as a kind of “translation” between two or more terminologies that aims to enhance the communication among people coming from different domains or expertise. The Visual Terminology Alignment tool (VisTA) enables the exact alignment for RDF/SKOS-like terminologies, suitable for integrating knowledge bases, rather than information retrieval systems. The tool provides a simple and friendly web-based user interface for the alignment between two terminologies, while it visualizes the terminology hierarchies, enables the interactive alignment process, and presents the alignment result.

In particular, the terminologies are graphs that can be visualized as multi-rooted trees so that their internal hierarchy can be exposed to and easily visited by the alignment authors. The tool facilitates the terminology alignment process providing: a flexible workspace with a drag’n’drop mechanism to create correspondences between terms, an extremely helpful highlighting mechanism to enable the user locate the hierarchy of a term and to be aware of the status of an alignment, a semantics constraint mechanism to support meaningful alignments and finally, alternative ways to search for specific terms in the terminologies.

VisTA is a module that conforms to a respective component foreseen by the Synergy Reference Model¹, an initiative of the CIDOC CRM Special Interest Group², for manipulation of data provisioning and aggregation processes, aiming at defining the processes needed to be executed or maintained between a data provider (the source) and a data aggregator (the target) institution, such as a museum and Europeana³

2 The alignment problem in VisTA

In VisTA we deal with the *target-driven* alignment of two terminologies as an *asymmetric* process aiming to the *subordination* of a source to a target terminology. The process results in *n×m correspondences* among the terms of the two terminologies, whereas a *correspondence* is regarded as a direct association between two terms.

For the successful subordination of the source and the production of consistent knowledge, we assume three *principles*:

- Terminologies and the alignment result are *acyclic* graphs preserving the taxonomy *subsumption (is-a)*
- The *broader* and *narrow* relations are *symmetrically inversed*
- *Reconciliation of the source terminology* with the target but not the reverse

The alignment process results in the integration of different terminologies, under one target terminology which is considered as a core structure. The reconciliation of the structure of the source terminology is supported by the tool, in order to remove any subterms of the

¹ http://www.cidoc-crm.org/sites/default/files/SRM_v1.5.pdf

² <http://network.icom.museum/cidoc/working-groups/crm-special-interest-group/>

³ <https://www.europeana.eu/>

aligned source term that violate the preservation of subsumption. The extension of the target terminology to more specific terms is based on the *broader match* and *exact match* relations.

Constraints (rules) applied during the process prevent from undesirable situations in the alignment result. *Multiple-inheritance* of terms, i.e. a term can have multiple parents, and subhierarchy *overlaps* may occur being allowed unless they break subsumption.

The alignment result produced in VisTA is a native RDF/SKOS graph that interconnects the two terminology graphs. The searching capabilities are really empowered, as the users of different terminologies are enabled to make queries using the common target vocabulary together with their own familiar vocabularies, to find more resources in their results.

3 General features of the system

3.1 Basic operations

The main operations provided by the tool comprise the editing and viewing of alignment correspondences between two terminologies. Also, the tool offers an easy to use interface to enable users manipulate the data involved in the alignment activity, such as import, export and delete terminology and alignment named graphs.

3.2 Alignment process

A single term of the source terminology can be aligned to multiple terms of the target terminology and vice versa. The process results in the subordination of the source to the target terminology utilising two fundamental hierarchy relations of the SKOS schema:

- a. skos:broaderMatch, when a source term becomes a narrow term of the target
- b. skos:exactMatch, when a source term is the same, with equivalent meaning, as a target term

3.3 Visualisation of the terminologies

The representation of terminologies as tree views (indented lists) enables accessibility, comparability and association between hierarchical structures. The tree view hierarchy is based on specific RDF relations, predefined in the main widget template, implying the hierarchy relation between broader and narrow terms. The term-nodes of a tree can be expanded/collapsed, selected and dragged for specific purposes of editing and viewing. When hierarchy relation is missing from a terminology then a flat list of the terms is presented.

3.3.1 Preferred languages

The preferred languages for displaying terminologies can be defined at the main widget template in the *preferredLangs* parameter. The RDF languages included in this value, as a comma-delimited list, are considered for filtering the requested languages. If no language is assigned that label is filtered out. Labels without a language assignment are filtered out. To

include those as candidate labels for displaying a term, it is required to include the empty value "" in the preferred language list.

3.4 Widget template configuration

The application is based on the *widget template* mechanism of the Metaphactory platform. The starting page, we propose the name *forth:term-alignment* for that, contains the required configuration used by the internal structures of the application. The given parameters can be modified to extend the operability to more terminology cases as the use of RDF relations cannot be foreseen in new terminologies. The parameters of the template are described below.

Parameter	Description
<i>exactMatchAlignmentRelation</i>	The RDF <i>exact-match</i> relation used in the Alignment process, as a structure of { title, iri }
<i>broaderMatchAlignmentRelation</i>	The RDF <i>broader-match</i> relation used in the Alignment process, as a structure of { title, iri }
<i>broaderAlignmentRelation</i>	The RDF <i>broader</i> relation used in the hierarchy of the source terms into the Alignment graph, as a structure of { title, iri }
<i>terminologyPrefices</i>	Optional. The prefices related to specific terminology iris, used with the labels of the terms, defined as a mapped list {"terminology-iri": "prefix", ...}
<i>alignedGraphNS</i>	The common namespace of the alignment graphs
<i>prefices</i>	The SPARQL prefices section with namespaces involving terminologies and schemas required for Alignment
<i>infoRelations</i>	SPARQL property path related to the requested information about a term
<i>equalityRelation</i>	SPARQL property path for relating equal terms (not for alignment matching)
<i>typeRelation</i>	SPARQL property path for requesting the <i>type</i> of a root term. Used in the <i>rootsQuery</i> parameter
<i>typesRootQuery</i>	SPARQL filtering expression with the types of a root term. Used in the <i>rootsQuery</i> parameter
<i>typesChildrenQuery</i>	SPARQL filtering expression with the types of a child term. Used in the <i>childrenQuery</i> parameter
<i>typesParentQuery</i>	SPARQL filtering expression with the types of a parent term. Used in the <i>parentsQuery</i> parameter
<i>hierarchyRelationOptions</i>	The list of RDF properties for selecting the desired hierarchy relation between a term and a subterm. The selected one is used in the parametrized queries of the template
<i>typesSearchQuery</i>	The RDF values of the types of a searched term. Used in the <i>searchQuery</i> parameter
<i>labelRelation</i>	SPARQL property path for querying the labels of the terms. Used in the parametrized queries of the template
<i>preferredLangs</i>	Optional. A list with the preferred RDF language values
<i>dropdownlimit</i>	Optional. The amount of rows in the dropdown lists for choosing terminologies for alignment. Default value is 10
<i>queries</i>	The set of parametrized queries of the template comprising <i>rootsQuery</i> , <i>childrenQuery</i> , <i>searchQuery</i>
<i>rootsQuery</i>	The SPARQL query template for retrieving the root terms of a

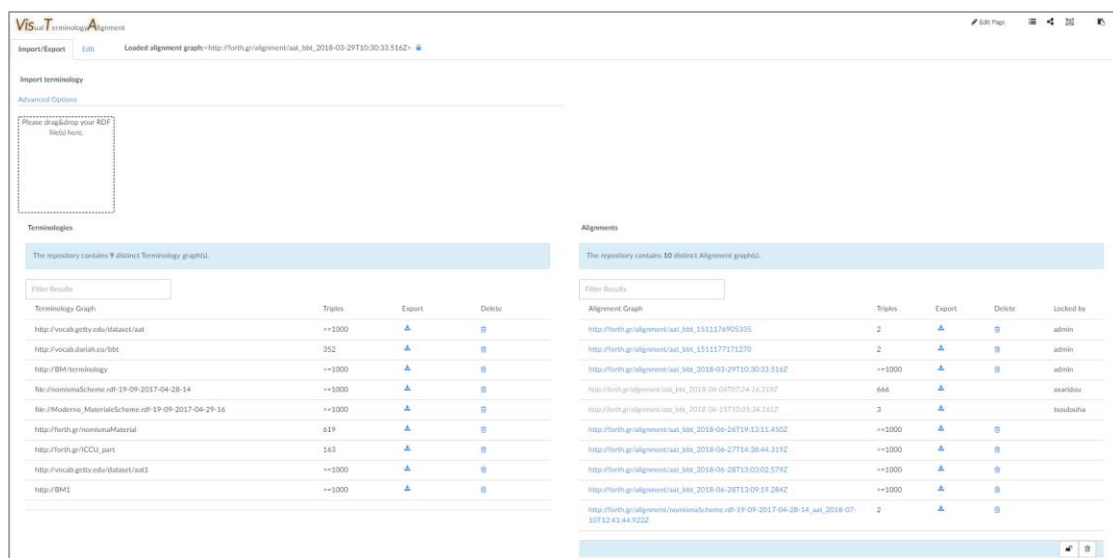
	terminology tree
<i>childrenQuery</i>	The SPARQL query template for retrieving the children of a term in a terminology tree
<i>parentsQuery</i>	The SPARQL query template for retrieving the parents of a term in a terminology tree
<i>searchQuery</i>	The SPARQL query template for performing a search in a terminology tree

4 User interface and functions

An easy to use interface helps the user to manage the Alignment process and result. Two tabs on top of the page transfer the user to the respective working area: *Import/Export*, to manipulate terminology and alignment graphs and *Edit*, to proceed with an Alignment process.

4.1 Import/Export: managing datasets

This page provides the appropriate options to manage the import and export of terminology data. Terminology data are distinct named graphs of SKOS/RDF data that can be used as source and target terminologies in an Alignment process. Although the Metaphactory system already has an import and export mechanism that can be utilised, this same mechanism is also embedded in the tool interface to provide quick access to data management and also make the tool an independent application.



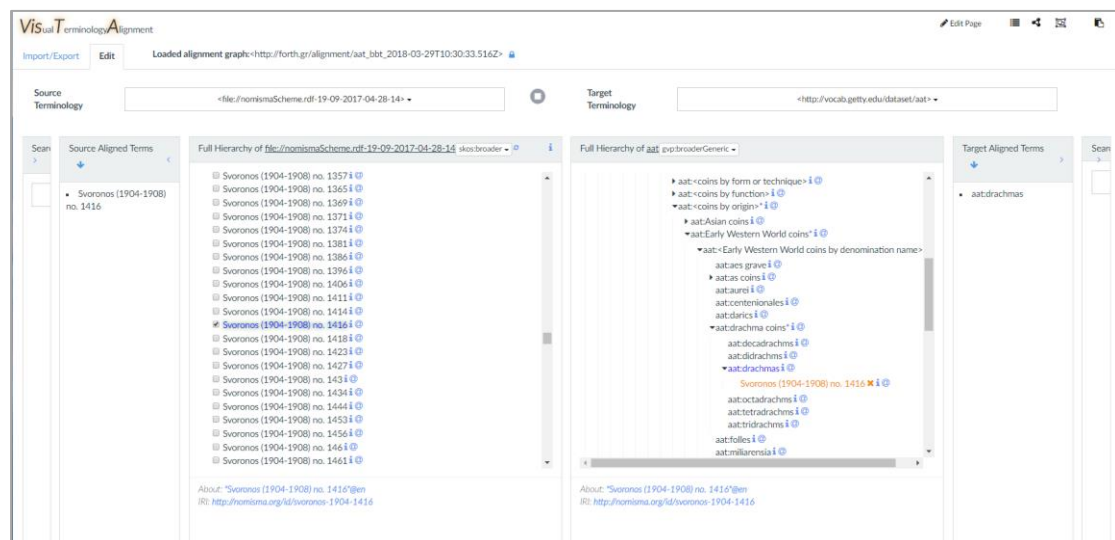
The “Import/Export” page

The source and target terminologies should be imported as RDF files. The *Import* mechanism enables the user to locate or drag ‘n’ drop SKOS/RDF files to upload the terminology data. The selected file will be imported to the graph store in the context of a named graph IRI, that is either auto-created or given by the user. The mechanism automatically detects the format of the given files in order to parse them properly and ingest them to the graph store.

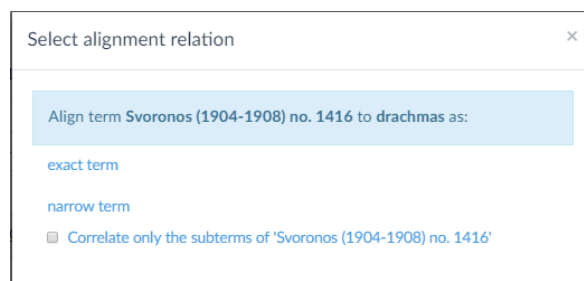
The *Export* mechanism helps the user to view, *download* and *delete* the related named graphs. The existing *Alignment* and *Terminology* named graphs are displayed in tables, where each row provides information about their size, their lock state and the appropriate options to delete a graph or download in the desired RDF format. Particularly the alignment graphs are displayed as links in order to be directly loadable for further editing. The locked entries by other users are disabled and can be available only to their owners.

4.2 Edit: loading/creating/updating alignments

The editing of an Alignment is the directed matching process performed from source to target terms. In the *Edit* page the user can choose the *source* and *target terminologies* in order to proceed with a new Alignment. The Alignment process comprises distinct steps of correspondences creation. On each step a dialog popup appears asking the user which type of relation to use for the currently aligned source and target term, prompting to *align a source to a target term as: exact or narrow term*. With the latter option the user can also enable the option for correlating only the subterms of the selected source term, as narrow terms to the target.




The “Edit” page



The “Select alignment relation” popup during editing

4.2.1 Loading terminology trees

The available terminologies of the graph store can be loaded either as the source or target object of an Alignment process, selected in the left or right dropdown list respectively. When


both the source and the target objects are selected, then the Alignment button  becomes enabled to initiate a new Alignment process. Also, an existing stored alignment can be loaded for editing by clicking on a graph IRI in the Alignments table of the Import/Export page.

When editing an Alignment, each terminology appears one next to the other so that the user can easily perform the creation of correspondences between them. The hierarchies provide the selection of the currently used hierarchy relation in order to display the terminology trees.

4.2.2 Searchable trees

Two input boxes appear in the outer left and right sides of the UI next to the source and target terminologies respectively, in the *Edit* page. They provide searching capabilities that return a filtered view of the searched terminology. Selecting terms from the filtered result they get properly highlighted in the original terminology view and the user can easily realize the absolute location of the terms into the terminology hierarchy. The searching options are presented in section [Searching for terms](#).





4.2.3 Representation of the aligned terms

After the relation is defined by the user the aligned source term appears on the target tree. Both the terms at the original trees are getting highlighted visualizing thus the new state and any parents of them appear with a start  signing thus that they contain an aligned sub-term. The way the alignment is visualized on the target tree depends on the applied relation:

- If the source term is a *narrow* match to the target term, a new node for the source term is created and added to the children of the target term, notating this way that it becomes a narrow term of the target. The new node may be a leaf or a branch containing the sub-hierarchy of the selected nodes from the source tree.
- If the source term is an *exact* match to the target term, then the source term appears on the same node with the target term and its child nodes are added to the children of the target term. This representation notates that the aligned terms are the same and hence they can share the union of their narrow terms.

The same source term can be aligned multiple times. The user can modify the internal hierarchy of a selected source term any time an alignment is performed, by excluding any of the existing or re-including some of its children. The most *recent state* of an aligned source term affects all its aligned instances on the target tree, synchronizing their internal hierarchies.

4.2.3.1 Buttons and symbols next to a term

	Flags terms that contain aligned sub-terms
	The button for the viewing the related info e.g. the scope notes of a term
	The button for viewing the location of the term in the hierarchy
	The button to remove an aligned source term from the target

4.2.4 Collapsible, selectable, excludable terms

The terms in the tree view, appear as *expandable/collapsible* and *selectable* nodes. The appearance of the short arrow ► on the left of a term node indicates that the node is a branch-term that can be expanded (and collapsed) further into narrow terms. A checkbox precedes the term label on every node making them selectable. Selecting a branch-term all its narrow-terms (child-nodes) are automatically selected creating thus a *tree-selection*. The current selection can be *cancelled* by either selecting a new non-selected term of the tree or deselecting the top selected term of the current tree-selection. Moreover, the user is given the ability to *unselect* terms from the selected hierarchy *excluding* this way any narrow terms from the alignment of a term.

4.2.5 Draggable terms

To create a correspondence between two terms first the source term has to be selected. A selected source term always becomes *draggable* and *droppable* on any target term in order to install a correspondence of the Alignment process. The drag-able terms may consist of a single term or a hierarchy of terms. On drop the dialog box *Select alignment relation* appears asking the user to decide whether the source term is an *exact* or a *narrow* term to the target.

4.2.6 Viewing the alignment correspondences

The panels labeled as *Source Aligned Terms* and *Target Aligned Terms*, in the middle of the source and the target areas respectively, contain sortable lists of the aligned terms of each terminology. Clicking on a term, in either list, the whole correspondence across the lists and the hierarchies gets highlighted. Particularly, the aligned terms are located in their original terminologies by highlighting the entire path of nodes from the root nodes up to their real locations.

4.2.7 Highlighted terms – Notation

The terms are highlighted depending on their state. At the source and target terminology trees the aligned terms and their children (narrow terms) are highlighted to indicate whether they are aligned as well as to visualize specific alignment information. The highlighting notation is presented below:

Terms style notation	Meaning
no-style	The term is not aligned neither explicitly nor implicitly
blue font color	The term is explicitly aligned (at both trees)
blue font color - italic	The term is implicitly aligned as a child of an aligned term (at the source tree)
orange font color	The term is an exact-match-source-term (at the target tree)
orange font color -italic	The term is a narrow-source-term (at the target tree)
orange text-shadow	The term is a child of the pointed exact-match-source-term (at

	the target tree)
<u>grey-blue dotted underline</u>	The term is a broader term of the one returned in the search result
<u>cyan solid underline</u>	The term is the one returned in the search result
lightpink dotted outline	The term is a broader term of the one selected by @
pink solid outline	The term is the one selected by @
blue solid radius border	A set of alignment correspondences is highlighted across the 4 panels: <i>Source Aligned Terms</i> , <i>source Full Hierarchy</i> , <i>target Full Hierarchy</i> , <i>Target Aligned Terms</i>
blue dotted radius border	The term is a broader term of the currently selected/viewed aligned term
green text-shadow	The term is the current source/target for the new correspondence

4.2.8 Deleting alignment correspondences

Next to the aligned source terms the ✖ option appears enabling the deletion of the current alignment correspondence. After deleting a correspondence the related source term and its children are removed from the target tree. Finally, the involved terms at source and target terminology trees get de-highlighted (unless there are more alignments for these terms).

4.2.9 Refreshing source hierarchy and the alignment graph

In the *Source Aligned Terms* the Refresh symbol ↺ enables the cleaning of the Alignment graph from any “orphan” non-aligned source subhierarchies that have been left from other deleting operations or from modifying the structure of already aligned source terms. After refreshing, any terms appearing as implicitly aligned in the source hierarchy, may become synchronized with the real state of alignment.

4.2.10 Information about a term

Next to the terms the *info* button ⓘ appears to present the information related to the term in the *Information area* under each terminology tree. The retrieved content comes from a parametrized query that can be configured at the main widget template in the *infoRelations* parameter.

4.3 Searching for terms

When editing or viewing an alignment there is always the need to search for terms, especially when deep hierarchies are involved. The search mechanism used in the tool is the Metaphactory *semantic-tree-input* component applied on both the source and the target trees, configured to operate in two modes:

- Search by term-ID*. Given an IRI if a matching term is found then it is returned as a *search result*.

- b. *Search by Text*. Typing a string (longer than 2 characters) a *full-text-search* is performed in the textual information of the terms, such as labels, notes, etc. The matching terms are returned as selectable tree nodes in a dropdown area, where the user can view the context of the terms in their local hierarchy. The final *search result* is returned after checking the desired term-nodes and clicking on the *Select* button.
- c. *Search by drag 'n' drop*. Dragging a term from the source or the target terminology and dropping it on the target or the source search box respectively, a search by the term-ID is performed

After the search result is returned by the component it is highlighted in the original terminology that it is referred to. The query applied for search can be modified in the main widget template.

Finally, a particular way to find more locations of a term in both hierarchies is enabled by the @ button next to the term.

4.4 Alignment rules

On the Alignment process, when a new correspondence is to be created and after the source and target terms become known to the application, the tool performs condition checks, to inform the user about the current state of the terms and prevent from useless or inconsistent situations.

The rules, applied before a new correspondence is created, are:

- a. The source term must not already exists (implicitly or explicitly aligned) in the hierarchy of the target term. In this case the alignment is cancelled.
- b. The source term may contain already aligned descendants. In this case the user is informed and the alignment is allowed.

4.5 Multi-user operation

The tool can be used by multiple users that share the same graph store for their terminology alignments. The creation of multiple alignment instances for a particular pair of terminologies is allowed to all users. Thus when a pair of terminologies is selected in the *Edit* page, a new *Alignment* is created for the specific pair. The user can *lock* a new or an existing unlocked graph in order to edit that privately and protect it from the other users.

The users are allowed to *lock* an unlimited amount of graphs and become the unique owners enabled to *edit* and *unlock* them. They are also allowed to *delete* one-by-one any alignment graph that is not locked by another user. Furthermore, they are enabled to *delete massively* any alignment graph(s) that are not locked, removing thus at once all the idle entries. Finally, the user is enabled to *massively unlock* of all the owned graphs and make them available to all users.

In public repositories the recommended alignment creation policy, when many users are involved, is to avoid creating multiple alignment instances for a particular pair of

terminologies in order to prevent from semantic inconsistencies due to conflicts and mistakes of the different alignment authors.

5 Alignment graph

It is a named graph taking its name automatically from the involved terminologies considering their source/target roles. For example when terminologies *namespaceA:A* and *namespaceB:B* are to be aligned as source and target correspondingly their alignment graph is named as *alignmentNamespace:A_B_timestamp*.

Apart from naming the alignment graph with the names of the aligned terminologies A and B, an important RDF triple is included into the alignment graph:

namespaceA:A, alignmentNamespace:isAlignedTo, namespaceB:B

The above is helpful to find out the IRIs of the aligned terminologies and their source/target role.

The alignment graph contains the all the essential information of the matching terms between the two terminologies as well as the children hierarchy of the terms coming from the source hierarchy. The structure of the aligned terms coming from the target terminology is considered to be constant thus there is no need to establish their structure in the alignment graph. Hence for the alignment result to be useful in practice, e.g. in a search mechanism, the appropriate searching area would be the union of the alignment graph and the target terminology.

Annex 1 – The main widget template (Version 5)

```
<terminology-alignment-tool

exactMatchAlignmentRelation = '{ "title": "exact term", "iri":
"http://www.w3.org/2004/02/skos/core#exactMatch" }'
broaderMatchAlignmentRelation = '{ "title": "narrow term", "iri":
"http://www.w3.org/2004/02/skos/core#broaderMatch" }'
broaderAlignmentRelation = '{ "title": "narrow term", "iri":
"http://www.w3.org/2004/02/skos/core#broader" }'
terminologyPrefices = '{ "http://vocab.dariah.eu/bbt": "bbt",
"http://vocab.getty.edu/dataset/aat": "aat" }'
alignedGraphNS = 'http://forth.gr/alignment/'
prefices = '
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX gvp: <http://vocab.getty.edu/ontology#>
PREFIX xl: <http://www.w3.org/2008/05/skos-xl#>
PREFIX bbt: <http://vocab.dariah.eu/bbt/>
PREFIX bds: <http://www.bigdata.com/rdf/search#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX hint: <http://www.bigdata.com/queryHints#>
'

infoRelations = '( (skos:scopeNote/rdf:value) | skos:scopeNote )'
equalityRelation = '( skos:exactMatch )'

typeRelation = '(skos:topConceptOf | a )'
typesRootQuery = '(?type = gvp:Facet || ?type =
<http://vocab.dariah.eu/bbt/ConceptScheme/Backbone_Thesaurus> ||
?type = skos:Concept)'
typesChildrenQuery = '(?type = gvp:Concept || ?type = gvp:GuideTerm
|| ?type = gvp:Hierarchy || ?type = gvp:Facet || ?type =
skos:Concept)'
typesParentQuery = '(?type = gvp:Concept || ?type = gvp:GuideTerm ||
?type = gvp:Hierarchy || ?type = gvp:Facet || ?type = skos:Concept ||
?type = bbt:Facet)'
hierarchyRelationOptions = 'skos:broader, gvp:broader,
skos:broaderTransitive, gvp:broaderGeneric, skos:narrower,
^skos:narrower'
typesSearchQuery = 'skos:Concept gvp:Concept gvp:GuideTerm
gvp:Hierarchy gvp:Facet'
labelRelation = '(skos:prefLabel | (gvp:prefLabelGVP/xl:literalForm)
| rdfs:label | dc:title)'
preferredLangs = '["el", "en", "en-us", "es", "it", "de", ""]'
dropdownlimit = '5'

queries = '
{

    "rootsQuery": "__prefices__

    SELECT DISTINCT ?item ?label ?hasChildren
    /*FROM __alignmentGraph__*/
    FROM __graph__
    WHERE {
        {
            ?item __typeRelation__ ?type .
            FILTER ( __typesRootQuery__ )
        }
    }
    }
}
```

```

        MINUS { ?item __parentshipRelation__ ?parent }
        OPTIONAL { ?item __labelRelation__ ?labelX. }
        BIND      (      IF(CONTAINS(STR(?labelX),\"unauthorised\"),
STR(?item) , ?labelX) as ?label )
    }
    OPTIONAL{
        {
            ?child __parentshipRelation__ ?item .
        }
        /*union{ GRAPH __alignmentGraph__{
            ?sourceT __exactMatchAlignmentRelation__ ?item .
            ?child __broaderAlignmentRelation__ ?sourceT . }
        }
        union{ GRAPH __alignmentGraph__{
            ?child __broaderMatchAlignmentRelation__ ?item .
        }
    }
    }*/
    }
    BIND (bound(?child) as ?hasChildren)

    __preferredLangs__

} ORDER BY ?label",

"childrenQuery": "__prefices__

SELECT DISTINCT ?item ?label ?hasChildren
FROM __graph__
WHERE {
    {?item __parentshipRelation__ ?parent . }
    /*union{ GRAPH __alignmentGraph__{
        ?item __broaderMatchAlignmentRelation__ ?parent .
    }
    }
    union{ GRAPH __alignmentGraph__{
        ?item __broaderAlignmentRelation__ ?parent . }
    }
    union{ GRAPH __alignmentGraph__{
        ?x __exactMatchAlignmentRelation__ ?parent .
        ?item __broaderAlignmentRelation__ ?x . }
    }*/
    /*OPTIONAL {graph __sourceGraph__ { ?item
__labelRelation__ ?labelX } . }*/
    OPTIONAL { ?item __labelRelation__ ?labelX }
    OPTIONAL {
        ?item __typeRelation__ ?type .
        filter (__typesChildrenQuery__)
    }

    OPTIONAL {
        {
            ?child __parentshipRelation__ ?item. }
        /*union{ GRAPH __alignmentGraph__{
            ?child __broaderAlignmentRelation__ ?item . } }
        union{ GRAPH __alignmentGraph__{
            ?child __broaderMatchAlignmentRelation__ ?item. } }
        union{ GRAPH __alignmentGraph__{ ?y skos:exactMatch ?item.
?child skos:broader ?y. } }*/
    }
    BIND (BOUND(?child) AS ?hasChildren)

```



```
        BIND (0.6 as ?score) .
    }
    BIND(bound(?child) as ?hasChildren)
    } ORDER BY ?score LIMIT 100"
}'
>
</terminology-alignment-tool>
```