RUBRICA – Manual

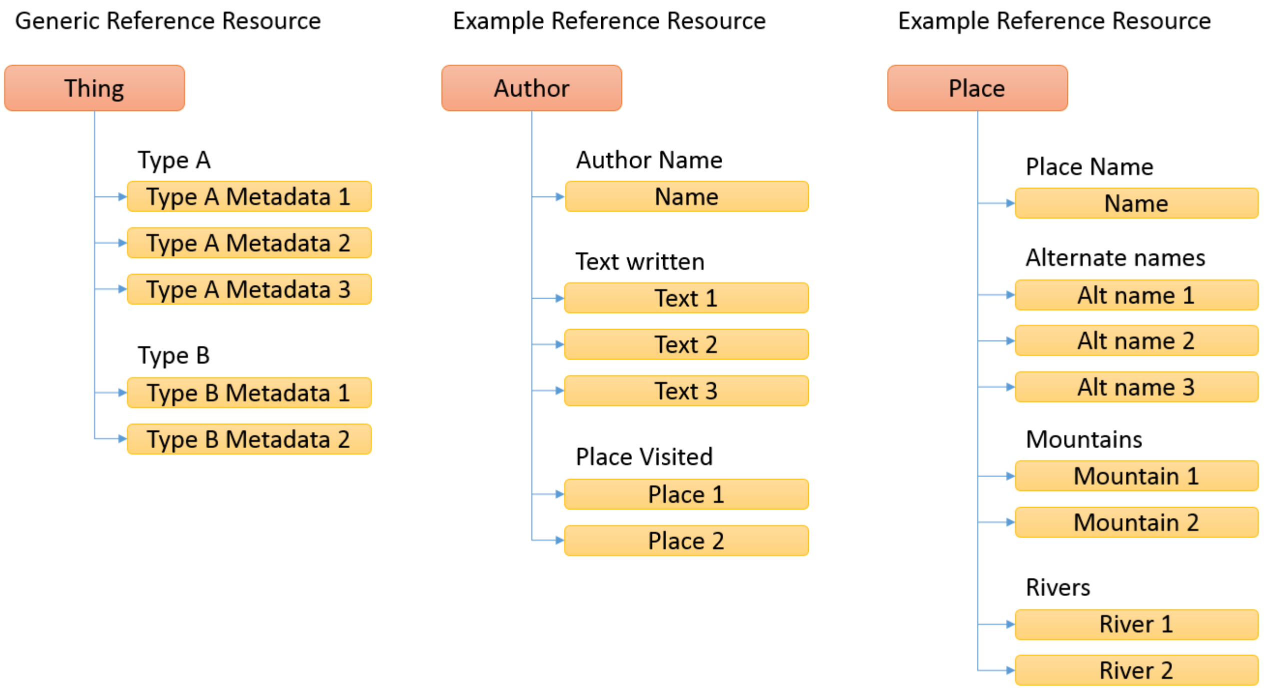
WHAT IS RUBRICA

RUBRICA (Reference Resources Integration plAtform), is being developed Within the PARTHENOS project. RUBRICA aims to foster the interoperability and integration of various reference resources used in different disciplines. Starting from trusted knowledge bases (i.e.: databases, thesauri, authority lists etc.) researchers could create, merge, edit and reuse specialized reference resources, developed according to specific research purposes, without performing repetitive tasks on each resource. RUBRICA also allows to share this knowledge base with other users thanks to the IT structure on which PARTHENOS is based.

WHAT IS A REFERENCE RESOURCE

In Arts and Humanities disciplines, a researcher working on a specific study needs to rely on reference resources, that help him developing his research. Reference resources can be seen also as an optimal place to start doing research because in addition to providing authoritative data on a specific topic, they summarize clearly useful information in an organized way.

From a slightly more technical point of view useful for the development of this manual the RR can be seen as lists of names/entity with aggregated metadata (metadata can have also different types).



Examples of Reference Resource structures

Following the evolution of the semantic web in which anyone can contribute to the creation of extraordinary knowledge graphs, RUBRICA produces and operates on RDF resources. A common ontology is required in order to provide convergent data, so to have subclasses describing a specific topic under the same taxonomic name. In the RUBRICA environment each resource should be mapped on the CIDOC Conceptual Reference Model (CRM) common semantic framework. In case the initial reference resources are not already modeled on a common model, the required mapping is made possible thanks to the already operative and user friendly service provided by PARTHENOS: the 3M Mapping Tool.

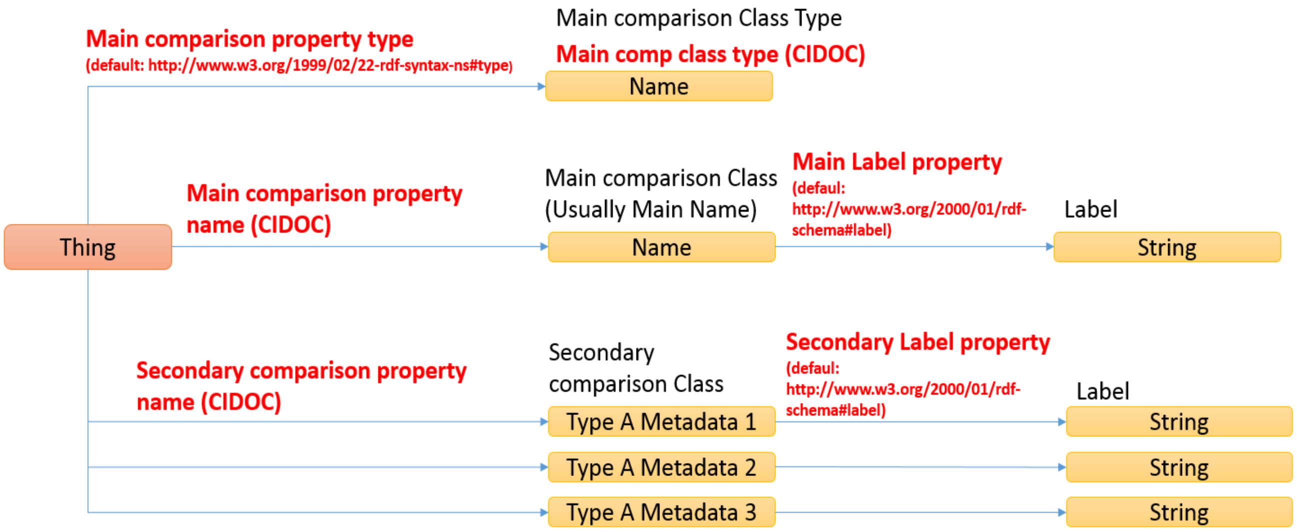
In the definition of the information in Triple RUBRICA uses the names of the properties and classes used in the dataset as input paramenters to perform its own core algorithms.

INPUT PARAMENTERS

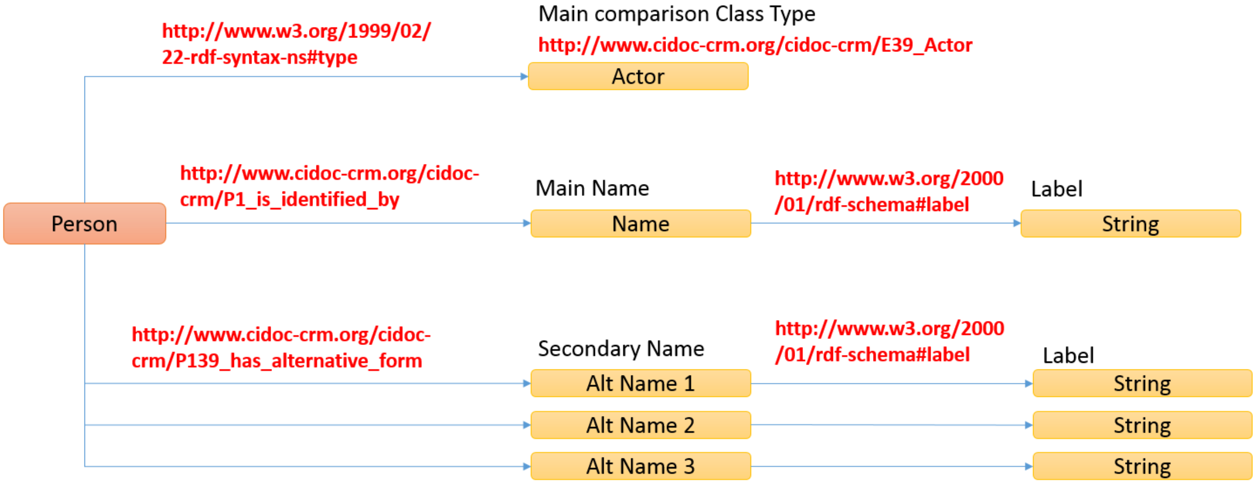
The user must initially define the names of the properties and classes that he wants RUBRICA to study for aggregation and specify it in the appropriate input Form. The tool works mainly on two comparison classes for each Reference Resource, one main and one secondary, using queries (created by a query building algorithm that uses input parameters), looking for common values to study (and eventually aggregate) between the different Reference Resources.

In order to facilitate the user in the process of inserting the parameters which initially may seem complicated, correct values were introduced by default, both class and property.

Below you will find in red the parameters required by RUBRICA and actual example of them.


Main comparison Class Type
Main comparison property type
(default: http://www.w3.org/1999/02/22-rdf-syntax-ns#type)
Main comp class type (CIDOC)
Name
Main comparison property name (CIDOC)
Main comparison Class (Usually Main Name)
Main Label property
(defaul: http://www.w3.org/2000/01/rdf-schema#label)
Label
Thing
Name
String
Secondary comparison property name (CIDOC)
Secondary comparison Class
Secondary Label property
(defaul: http://www.w3.org/2000/01/rdf-schema#label)
Label
Type A Metadata 1
String
Type A Metadata 2
String
Type A Metadata 3
String

General schema parameters

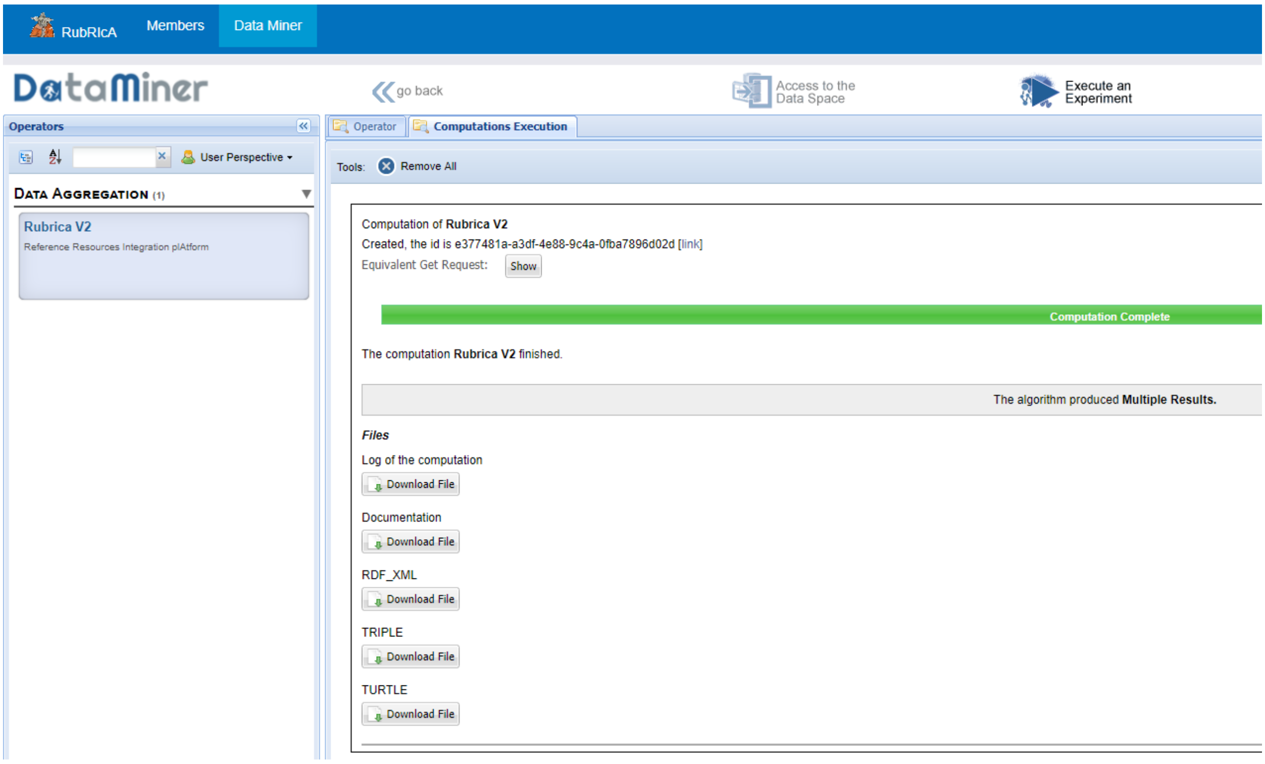

http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Main comparison Class Type
http://www.cidoc-crm.org/cidoc-crm/E39_Actor
Actor
http://www.cidoc-crm.org/cidoc-crm/P1_is_identified_by
Main Name
http://www.w3.org/2000/01/rdf-schema#label
Label
Person
Name
String
http://www.cidoc-crm.org/cidoc-crm/P139_has_alternative_form
Secondary Name
http://www.w3.org/2000/01/rdf-schema#label
Label
Alt Name 1
String
Alt Name 2
String
Alt Name 3
String

Author type schema parameters, in red the parameters used by RUBRICA to aggregate data

In addition to these parameters, the names of the Reference Resources and the XML files of their databases must also be provided.

OUTPUT

RUBRICA provides 5 files and a link as output: The first is a log file provided directly by the DataMiner of the D4Science platform useful for observing any errors that occurred during the process. The second file is a documentation file describing the various steps in RUBRICA that were used to aggregate the data. The documentation also includes a brief summary of the aggregate data. It also provides queries to be used on the final dataset created to simplify the work of the researcher who can use these queries to refine the searches within the dataset. The other three files are the same final aggregation database but in a different format (RDF-XML, TTL and Triple), which may be useful for interoperating data in data management platforms such as triple stores (Eg: Virtuoso). Besides these files, like all the algorithms integrated in the D4Science DataMiner, it is possible to use the link provided at the end of the process to replicate the experiment.



Computation of **Rubrica V2**

Created, the id is e377481a-a3df-4e88-9c4a-0fba7896d02d [link]

The computation **Rubrica V2** finished.

The algorithm produced **Multiple Results.**

DATA GROUPING AND EXPECTED RESULTS

RUBRICA has a 4-Phases data grouping method.

First phase of data grouping: Comparison of a main Entity label of the first Reference Resource with a main Entity labels of the second Reference Resource.

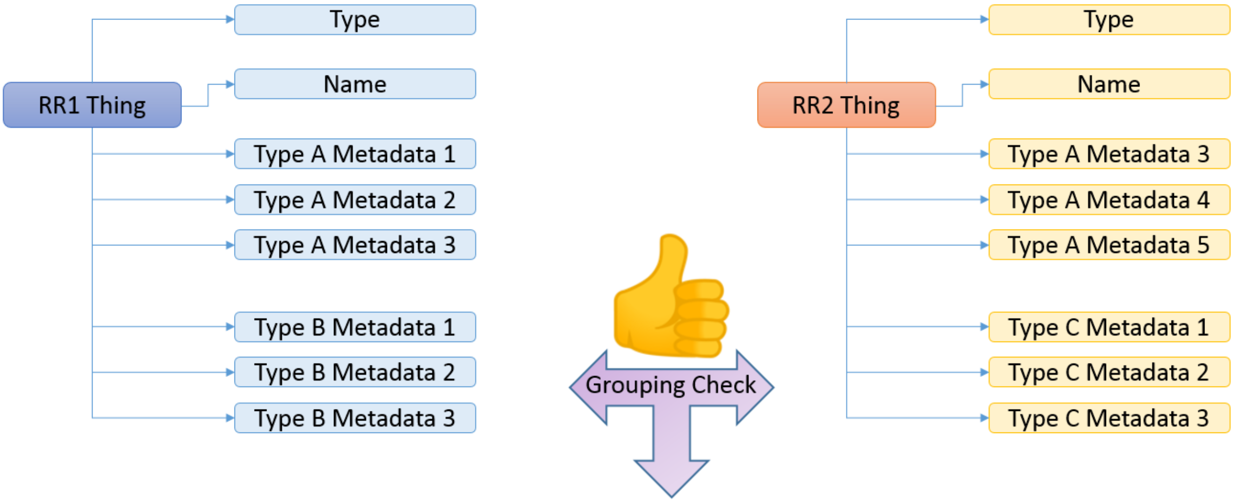
Second phase of data grouping: Comparison of a main Entity labels of the first Reference Resource with all the variant Entity labels of the second Reference Resource.

Third phase of data grouping: Comparison of all the variant names of the first Reference Resource with all the main names of the second Reference Resource.

Fourth phase of data grouping: Comparison of all variant Entity labels of both the Reference Resources. Data will be merged if at least 4 variant Entity labels will be found.

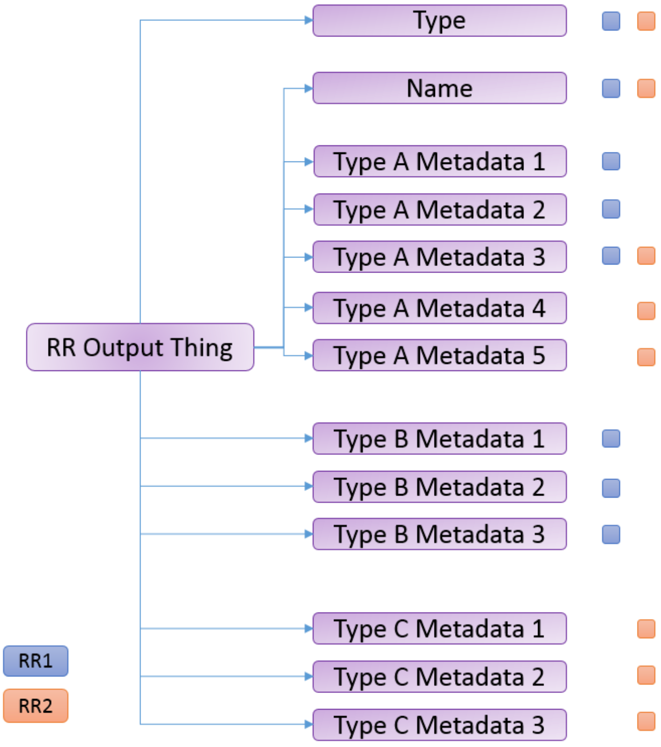
Each Data grouping phase result is described in detail in the documentation file.

The application aggregates the various entities updating the classes and joining the various properties if two entities are found to be equal.



Example of database structure of same taxonomical argument but different classes

This makes it clear that the final database will have aggregate classes each with the sum of the properties and classes of the entities that turned out to be aggregated.



Resulting output example data schema

In order to maintain a simple data navigation within the RUBRICA outputs, the application adds as an explicit RDF note the origin (name of the Reference Resource) of each entity present in the Database, specifying if it has been aggregated during the grouping process.