ModRef Project: Data Migration into CIDOC-CRM Triplestores and Factorisation

Pascaline Laure Tchienehom

Université de Paris 10 - Labex "Les passés dans le présent"
Nanterre, France
Email: pkenfack@u-paris10.fr

***Abstract*—ModRef is a project from the laboratory Labex "Les passés dans le présent", which coordinates various projects on digital humanities. ModRef focuses more precisely on the semantic web and linked open data. The goal is to move heterogeneous data into triplestores also called data warehouses or collections of RDF files in order to improve the sharing, exchange and discovery of new knowledge. For this purpose, the CIDOC-CRM norm has been chosen since it is, at present, the reference for the semantic description of museographic data or cultural heritage data. In order to realise the proof of concept of ModRef, a general architecture has been defined, a semantic modelling and data mapping of selected sub-projects of ModRef have been proposed, triplestores have also been created. A web application has been implemented and deployed. This web application describes the ModRef project, as well as it enables visualising, querying and exploring created triplestores. We also present and discuss the simplification or factorisation of the migration of data into CIDOC-CRM triplestores.**

Keywords–Digital Humanities; Semantic Web; Linked Open Data; Triplestores; CIDOC-CRM.

I. Introduction

The laboratory Labex "Les passés dans le présent" accompanies several projects in Social and Human Sciences (SHS) on issues related to digital humanities [1][2]: from dematerialisation of data to their structural description and even to their semantic description as well. ModRef (Modelling, References and Digital Culture) is a project from Labex that gather together a set of sub-projects with the goal of migrating their data into triplestores or data warehouses or collections of RDF (Resource Description Framework) files in order to improve the sharing, exchange and discovery of new knowledge. For this purpose, the CIDOC-CRM (International Committee for DOCumentation-Conceptual Reference Model) norm [3] has been chosen because it is currently the reference for the semantic description of museographic data or cultural heritage data [4]. The aim is globally to move from non-structured or semi-structured data to structured data and afterwards from structured data to semantic data. The semantic web then provides a solution to perform these data migrations.

The semantic web [5][6] is not only a concept but also an architecture [7], which is increasingly used in several applications. The semantic web architecture is a set of independant layers that collaborate to perform various tasks. This architecture describes data from their representation to their exploitation by applications or semantic web agents. Hence, various norms of data representation exist. The CIDOC-CRM is an example of semantic norm and more precisely a conceptual reference model or ontology. The aim of semantics and also the aim of the various metadata languages or semantic norms that define semantics is to provide an homogenous framework for representing and querying heterogenous data in order to reduce information silence and therefore improve the discovery of knowledge. Hence, ModRef project aims at realising a migration of data towards CIDOC-CRM triplestores using core data originally from heterogenous data sources where heterogeneity is based on contents and initial logical structure (spreadsheets, relational databases, XML files) as well.

In this paper, we present the ModRef project through: a general description of the semantic web, the linked open data and the CIDOC-CRM norm, in Section II; the general architecture of the ModRef project, in Section III; a CIDOC-CRM semantic modelling and data mapping of the three pilot sub-projects of ModRef with the CIDOC-CRM graph, in Section IV; a migration of data into CIDOC-CRM triplestores with and without factorisation, in Section V; a visualisation and exploitation of triplestores through the web application [8] that has been developped and deployed, in Section VI; the evaluation procedure and results, in Section VII.

II. Semantic Web, Linked Open Data and CIDOC-CRM General Presentation

The semantic web architecture is made of different layers [7] (see Figure 1). Those layers can be grouped in several categories: representation, reasonning, querying, trust and interaction (of web applications). This architecture describes a set of norms that any semantic web application should meet.

Representation layers define a common socle for all other layers of the semantic web architecture. These layers describe the representation of addresses (URI, Unicode), logical structures (XML) and semantics of data (RDF).

Addressing guaranties a unique reference to every resource by using namespaces that define prefixes of URIs (generic types of URLs) for a set of data.

Structuring data allows to describe the organisation of descriptive elements of a given data. It then defines the logical structure of the data. There are different types of generic model [9] describing logical structures (description of data that shows its components):

1) *attribute-value or key-value or flat structure model*. Resources are described by a list of attribute-value pairs. This model is easy to manage though it has a drawback pertaining to the lack of structure for the representation of complex information such as hierarchies for instance and hence prevent from making complex analysis related to structure. In this

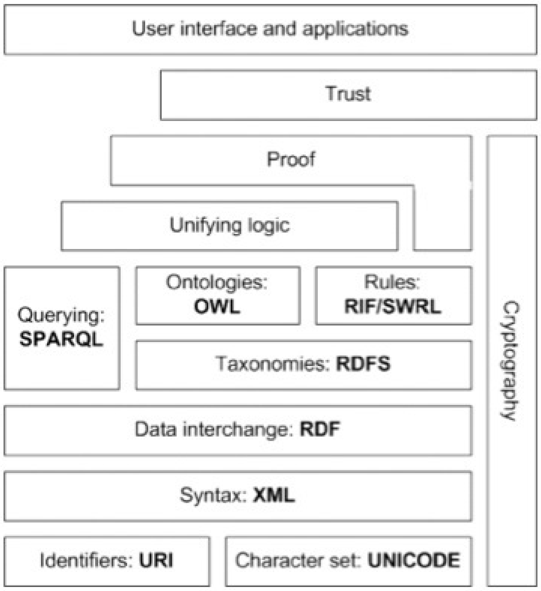


Figure 1. Semantic Web Architecture.

model, attributes are independant to each other for the attribute is considered as key. Consequently, it is forbidden to have two attributes with the same name since the name is the key of the descriptive element;

2) *hierarchical model* allows the structuration of logical structure, generally in the form of a tree like XML. This model allows a more precise description and analysis of data structure by creating composition links between nodes or attributes. Several nodes can then have the same name but different paths in the tree. XML is a standard for information exchange on internet. An XML document can describe databases or complex spreadsheets. There are also many XML norms such as: XML-EAD (Encoded Archival Description) for archives description, XML-TEI (Text Encoding Initiative) for text documents.

On the other hand, models based on logical structure are generally bound to the context or applications for which they have been defined and then are less re-usable. So, from one application to another, identical characteristics can be described by different logical structures. It will then be very difficult without a consensual semantics (metadata languages norms or standards) to query heterogenous data models.

Semantics of data is an answer to homogenise heterogeneous data sources. There are various data representation models based on semantics [10] [11] by using metadata languages that describe concepts and/or links between concepts (Dublin Core, RDF/RDFS/OWL, FOAF, Wordnet, CIDOC-CRM). These languages will enable more interoperability given the fact that their semantics is public and published through namespaces. Most of those metadata languages rely on RDF representation formalism which is an oriented graph with graph's paths define by a sequence of triples generally describe as follows: *[subject, predicate, object]* or *[domain, property, range]*. Note that various metadata languages can be used within the same semantic model and some of those metadata languages already define a conceptual model or ontology.

There are many standards for metadata languages. *RDFS and OWL* extend the *RDF* language and are more reasonning languages (use of properties such as: specialisation, generalisation -rdfs:subClassOf-, inverse -owl:inverseFunctionalProperty-, equivalence -owl:equivalentClass, owl:equivalentProperty-, identity -owl:sameAs-, symmetry, transitivity) of the semantic web whereas RDF is more of a representation language. In the same way, *FOAF* (Friend Of A Friend) describes people and organizations, what people do, links between people and is mainly used in social networks.

Wordnet is an ontology (formal descriptive model of the common vocabulary within a community in order to share information in a given domain), a lexical database developped by linguists of the laboratory of cognitive sciences of Princeton University (cf. http://wordnet.princeton.edu/wn20/). It aims at listing, classifying and linking in various ways the semantics and lexical contents of languages. There are different versions of Wordnet for different languages: english, french. Wordnet has been converted in RDF format. Note that lexiques, dictionnaries and thesaurus are used to construct ontologies and by that means contribute to the semantic description of data.

There are various data representation models based on semantics [10][11] that use metadata languages to describe concepts and/or links (properties or predicates) between concepts or instances of concepts. Those metadata languages are: Dublin Core, RDF, RDFS, OWL, FOAF, SKOS, Wordnet, CIDOC-CRM and so on.

The *CIDOC-CRM* [3] is a conceptual reference model for describing museographic data or cultural heritage data. The version of the CIDOC-CRM norm that we have worked with is the version 6.2 of may 2015. It describes 94 classes and 168 properties. In 2006, the CIDOC-CRM has become a norm ISO 21127:2006 but work on that norm has started since 1996. This norm describes general characteristics of objects (identifier, type, title, material, dimension, note) but also history of objects through events or activities (transfer of custody -former localisations, current localisation-, origin, discovery, curation, attribute assignement, measurement), as well as relations between objects or parts of objects (bibliography, composition, similarity, other representation -photo, drawing, painting-, inscription). An OWL implementation of the CIDOC-CRM by the University of Erlangen-Nuremberg is available [12] and the namespace of that implementation of CIDOC-CRM is usually prefixed by "*ecrm*".

The general structure of the CIDOC-CRM is described in Figure 2. The root class of all CIDOC-CRM entities is the class *E1 CRM Entity* and it is subdivided in direct sub-classes, among which the two main classes are:

1) *E77 Persistent Item*, which is the generic class of persistent entities. A persistent entity is an entity that can survive over an indeterminate time, such as: persons, objects, ideas, concepts. Those entities can have a beginning or an end of existence;

2) *E2 Temporal Entity*, which is the generic class of temporal entities. A temporal entity is an entity bounded by time (with a beginning and an end time), such as: event, beginning of existence, end of

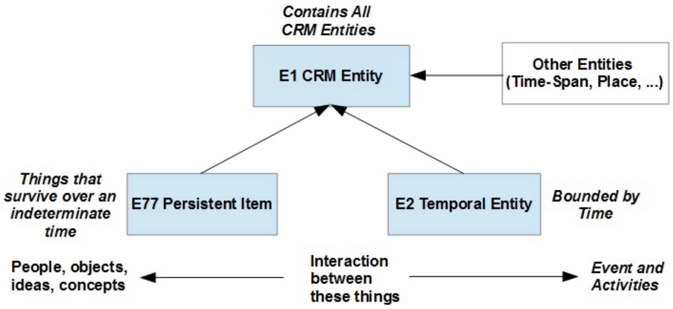

Contains All CRM Entities
E1 CRM Entity
Other Entities (Time-Span, Place, ...)
Things that survive over an indeterminate time
E77 Persistent Item
E2 Temporal Entity
Bounded by Time
People, objects, ideas, concepts
Interaction between these things
Event and Activities

Figure 2. General Structure of CIDOC-CRM Entities.

existence, activity, creation, production, modification, transfer of custody, curation, attribute assignement, measurement.

The other direct sub-classes of the root class *E1 CRM Entity* are the classes *E52 Time-Span, E53 Place, E54 Dimension, E92 Spacetime Volume*. In general, the CIDOC-CRM describes entities but also interactions that can exist between entities: interactions between persistent entities; interactions between temporal entities; interactions between persistent and temporal entities; general interactions between entities (for instance, interactions that exist between persistent or temporal entities and other general entities describing time-span, place, dimension). There also exists interactions between entities and primitive values (string, number, date time).

Besides, various projects around the world work on the migration of data into triplestores (CIDOC-CRM or not):

1) The *British Museum* [13], which is a museum on history and culture and that uses the CIDOC-CRM;
2) *Arches* [14], which is a collaboration between the Getty Conservation Institute (GCI) and the World Monuments Fund (WMF) on immovable cultural heritage (monuments, bridges) and that uses the CIDOC-CRM;
3) *DBPedia* [15], which is an online encyclopedia widely used [16] and that does not use the CIDOC-CRM norm but various metadata languages, such as: *dbpedia, foaf, umbel, schema.org, dublin core, geo*;
4) *Nakala* [17], which is an online service to upload, document and exhibit (museographic) data and that does not use the CIDOC-CRM norm but various metadata languages, such as: *foaf, skos, dublin core, vcard*.

The specificity of our web application is that it deals with heterogeneous data sources according to the contents and the logical structures (spreadsheets, relational databases, XML files) of data. Data migrated into triplestores are opened through our web application. This application provides a visualisation service of triplestores under three differents formats: *rdf, triples, attribute-value summary*. The web application also allows querying triplestores separately or together by using *"Endpoint Sparql"* (interface for typing and executing Sparql query, where Sparql is a querying language for RDF files) and *general query forms* that are useful for those who do not know the Sparql query language [18] and the CIDOC-CRM language.

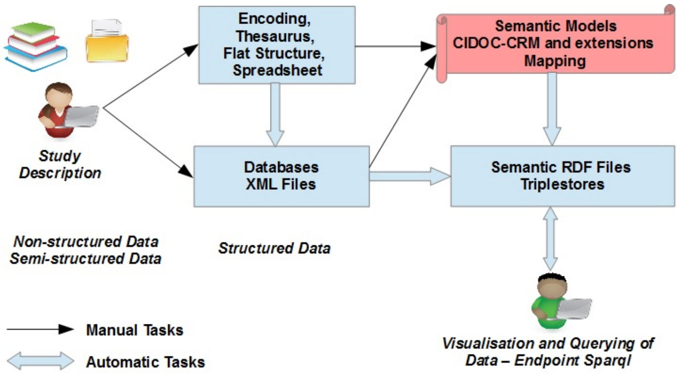

Encoding, Thesaurus, Flat Structure, Spreadsheet
Semantic Models CIDOC-CRM and extensions Mapping
Study Description
Databases XML Files
Semantic RDF Files Triplestores
Non-structured Data Semi-structured Data
Structured Data
Manual Tasks
Automatic Tasks
Visualisation and Querying of Data – Endpoint Sparql

Figure 3. Architecture of ModRef project.

III. Architecture of the ModRef project

The architecture of the ModRef project, illustrated in Figure 3, describes the various processes of data digitisation from the creation of digital data based on an expert knowledge for instance to the visualisation and querying of data by a user. Data can go through several transformations before being available in triplestores. Hence, we can move from non-structured or semi-structured data (notes, reports, books, web sites) to structured data described by logical structure. This logical structure can be a flat structure of the form *attribute-value* or *spreadsheet*, but it can also be more structured using *relational databases* or *XML files* that are, in our context, XML-EAD (Encoded Archival Description) files [19]. These various descriptions usually make use of thesaurus (controlled vocabulary of descriptive terms or not). From those structural descriptions, we can build a semantic description of data with a semantic RDF graph which relies on standards or norms. In our context, we have used the CIDOC-CRM norm to generate triplestores through a mapping between data and the CIDOC-CRM graph. These triplestores can then be used by semantic web applications or ”Endpoint Sparql”. The first stage of data transformation (from non-structured or semi-structured data towards structured data) is performed within each sub-project of ModRef whereas ModRef project itself focuses more on the second stage of data transformation (from structured data to semantic data).

Therefore, to realise the proof of concept (POC) of ModRef, three pilot projects have been selected:

1) *CDLI (Cuneiform Digital Library Initiative)*: Digital museum on antique documents in cuneiform writing [20];
2) *ObjMythArcheo*: corpus of antique archaeological objects with mythological iconography [21];
3) *BiblioNum*: Digital library on history of France during the 20th century [22].

Table I compares data of the three pilot projects of ModRef based on 5 criteria: descriptive texts size, number of objects, logical structure type, number of elements of the logical structure and data description language.

IV. ModRef CIDOC-CRM Modelling and Data Mapping

We have identified the useful CIDOC-CRM classes, for which at least one path leads to a non-null value, for the data

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. DATA COMPARISON.

	CDLI	**ObjMythArcheo**	**BiblioNum**
Texts size	300 Mo	100 Mo	100 Mo
Number of objects	313 332 objects	17 424 objects	77 collections - 62 392 files
Logical Structure	Database of type Spreadsheet	Relational Database	XML-EAD
Number of elements of the structure	1 table with 61 attributes	59 tables	146 XML-EAD elements
Language	English	French-English	French

modelling of our three pilot projects. This modelling represents extracts related to the four following themes or subjects:

1) general characteristics of objects (identifier, type, title, material, dimension, note or description), bibliography, composition and similarity of objects;
2) events of beginning of existence (origin) and end of existence;
3) miscellaneous activities (transfer of custody, attribute assignement, measurement);
4) inscriptions and other representations (photo, drawing, painting).

In general, those extracts are constant because with the CIDOC-CRM, it is possible to identify all potential paths that lead to a given information. A semantic graph is thus a set of nodes and oriented arrows that fulfill some constraints and rules (shortcut, entailment, inverse). These constraints and rules define the consistency and validity of the model.

In the following sections, we will describe the four different themes (graph's extracts) for the CIDOC-CRM modelling of our pilot projects and also an instance of data mapping with the corresponding CIDOC-CRM semantic graph snippet. Note that the mapping or alignment principle is globally the same for all themes and for all pilot projects.

A. Modelling of general characteristics

General characteristics of an object is defined more often by interactions through short graph's paths. Those characteristics describe for an object various information, such as: identifier, type (categorisation), title, material, dimension, note or general description.

The modelling of the general characteristics of objects of the ModRef project is illustrated in Figure 4. In this figure, there are two different graph's paths for defining the dimension of an object:

1) a *short path* or *shortcut path* that links class *E70 Thing* to class *E54 Dimension* with the property *P43 has dimension* by the triple *[E70 Thing, P43 has dimension, E54 Dimension]*;
2) a *long path* with more nodes to fill. This path is described by the following triples: *[E1 CRM Entity, P39i was measured by, E16 Measurement], [E16 Measurement, P40 observed dimension, E54 Dimension]*. With this path, we can fill more information related to the activity of measurement *E16 Measurement*. Actually, the class *E16 Measurement* is a type of activity because classes *E13 Attribute Assignment*, *E7 Activity* and *E5 Event* belong to its hierarchy (see Figure 5).

Besides, it is authorised to fill various paths leading to the same information in a CIDOC-CRM graph. Therefore, we sometimes have to choose between the different possibilities when we do not have the necessary information to describe a given path. This is the case mostly when a temporal entity is used in the path.

On the other hand, Figure 4 also illustrates other interactions between persistent entities, such as: *P70i is documented in* for bibliographic references, *P46 is composed of* for objects composition, *P130 shows features of* for objects similarity, *P128 carries* for relation between an object and an entity carried by or engraved on the described object, such as an inscription for example.

B. Modelling of events of beginning and end of existence

An important activity on museographic data is the description of their origin (beginning of existence) in order to define their date of origin, their place of origin and eventually the participants to their origin or creation. The modelling of beginning and end of existence events of objects in the ModRef project is illustrated in Figure 5. The CIDOC-CRM allows to define for each event three main information: date or period, place and participants.

For the beginning of existence (origin), we use the event *E63 Beginning of Existence* and the following patterns of triples: *[E77 Persistent Item, P92i was brought into existence by, E63 Beginning of Existence], [E2 Temporal Entity, P4 has time-span, E52 Time-Span], [E52 Time-Span, P78 is identified by, E49 Time Appellation], [E4 Period, P7 took place at, E53 Place], [E53 Place, P87 is identified by, E44 Place Appellation], [E5 Event, P11 had participant, E39 Actor], [E63 Beginning of Existence, rdfs : subClassOf, E5 Event], [E5 Event, rdfs : subClassOf, E4 Period], [E4 Period, rdfs : subClassOf, E2 Temporal Entity]*. Besides, for the beginning of existence (origin) we may also start from activities *E65 Creation* or *E12 Production*, which have as super-classes the classes *E63 Beginning of Existence* and *E7 Activity* (see Figure 5).

For the end of existence, we use the class *E64 End of Existence* or any of its sub-classes and we will then be able to define the date, the place and the participants to the end of existence of an object.

C. Modelling of miscellaneous activities

Figure 6 illustrates an extract of our model for the description of activities in general, and for the description of the activity *transfer of custody* in particular. Hence, to link an object to an activity of transfer of custody, we use the property *P30 transferred custody of* (or its inverse *P30i custody transferred through*) between the target activity (*E10 Transfer of Custody*) and the physical object (*E18 Physical Thing*). Moreover, for a transfer of custody, we can describe the various protagonists of the transfer (*P29 custody received by, P28 custody surrendered by*) and also describe eventually a history of the different transfers of custody related to a specific object or document. Note that there also exists a shortcut path that does not use the transfer of custody activity but that allows to define the current or former keepers or owners of an object (*P49 has former or current keeper, P50 has current keeper, P51 has former or current owner, P52 has current owner*).

Generally, for an event or an activity, we can describe the date (or period), the place and the participants (or actors)

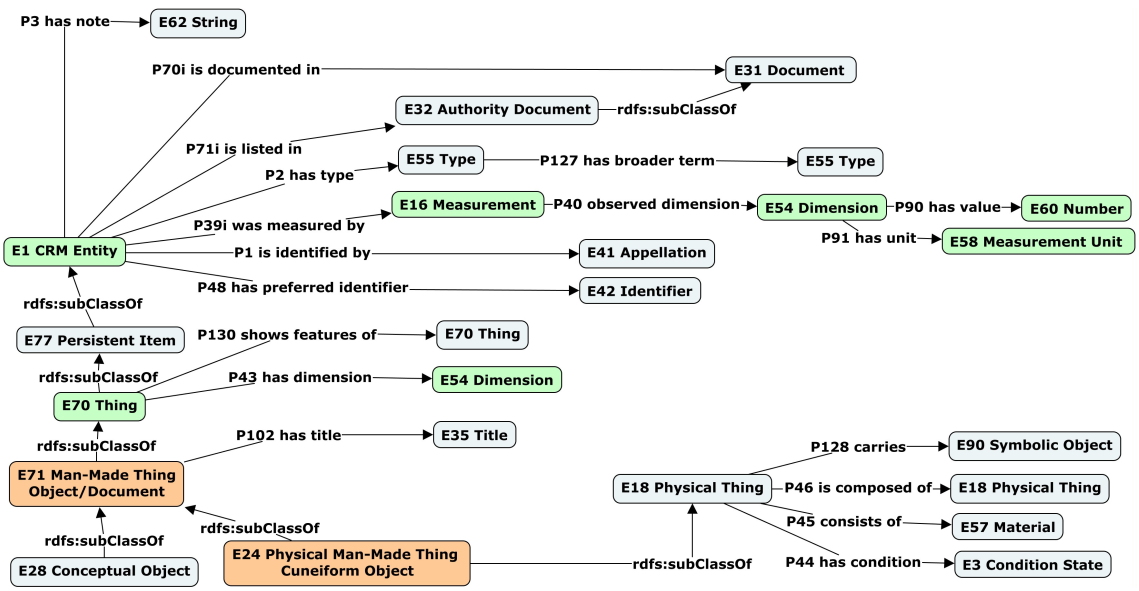


Figure 4. Modelling of general characteristics.

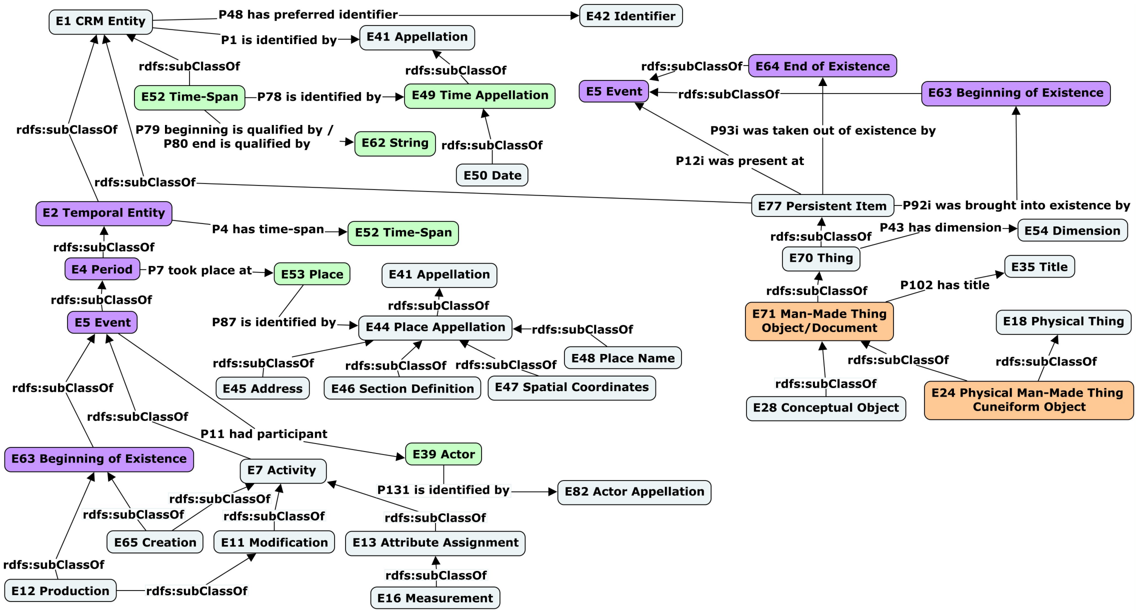


Figure 5. Modelling of events of beginning and end of existence.

of the event or activity. Specifically, for an activity (transfer of custody, attribute assignement, measurement, curation), we will be able to define more information, such as: the procedure or technique used (*P33 used specific technique, P32 used general technique*), the various objects used (*P16 used specific object, P125 used object of type*), the purpose of the activity (*P20 had specific purpose, P21 had general purpose*).

D. Modelling of inscriptions and other representations

Figure 7 illustrates a snippet of the model for the description of inscriptions engraved on objects or the description of other representations (photo, drawing, painting) of these objects (physical or conceptual). Hence, to link an object to its inscription, we use the property *P128 carries* between a physical object (*E18 Physical Thing*) and a symbolic objet (*E90 Symbolic Object*) that is inscribed on the target described object. This will then help to find, for instance, all objects engraved with a given inscription (seal, signature).

Besides, to link a photo or drawing or painting to an object (physical or conceptual), we can use a set of properties:

1) *P62 depicts* for describing the link between the photo or drawing or painting (here, *E24 Physical Man-Made Thing*) and the target entity *E1 CRM Entity* (physical or conceptual object) represented by the photo, drawing or painting. This property does not

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

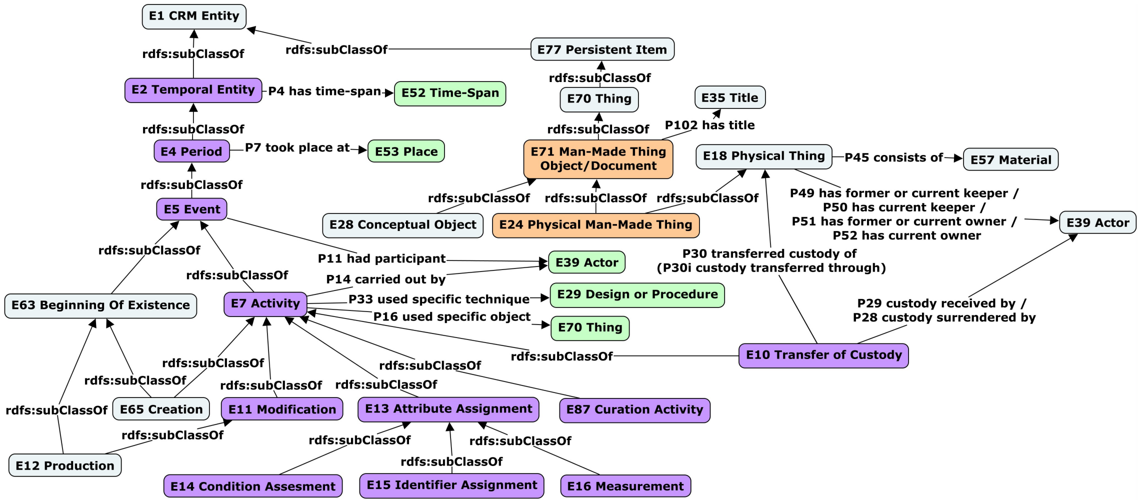


Figure 6. Modelling of miscellaneous activities.

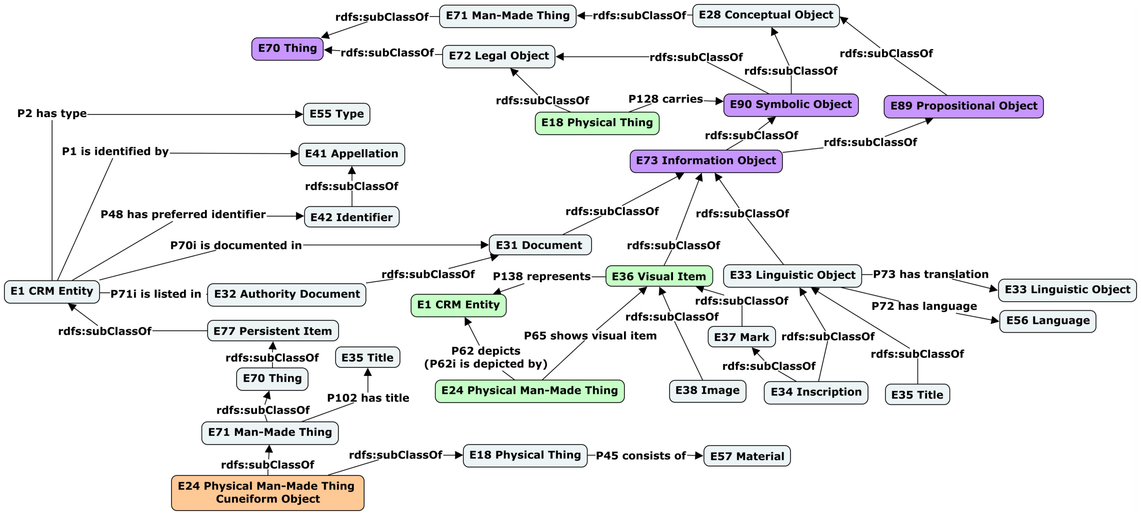


Figure 7. Modelling of inscriptions and other representations of an object.

describe inscriptions or other information engraved directly on objects;

2) *P65 shows visual item* that links the photo or drawing or painting to a visual representation (*E36 Visual item*) of the described object represented by the photo or drawing or painting;

3) *P138 represents* that links a visual representation (*E36 Visual item*) of an object to the target described object (*E1 CRM Entity*).

Note that the property *P62 depicts* is a shortcut of properties *P65 shows visual item* and *P138 represents*. The photo or drawing or painting is usually described with the class *E24 Physical Man-Made Thing*.

E. Data Mapping or Alignment

Data migration into triplestores requires an alignment stage of data with extracts of the CIDOC-CRM graph proposed. This mapping is essential due to the heterogeneity of initial data structured description and also due to the diversity of pilot projects. This matching is not properly a programmatic task but relies on logical structure details specific to the structural descriptive model chosen by each sub-project. It is a task halfway between modelling and programming that it enables to oversee more clearly. Note that this task should not be confused with the alignment between ontologies or owl/rdf files [23][24] since here it is rather an alignment between the CIDOC-CRM ontology and raw data from databases or XML files (in our context, XML-EAD files).

Primarily, the mapping consists in filling a semantic graph's nodes. Terminal nodes will be filled with values extracted from data logical structures of the corresponding sub-projects whereas non-terminal or intermediary nodes will be filled with URIs and will then define by that means paths to terminal nodes. Note that a particular attention must be drawn to the construction of URIs for their readability, as well as for the consistency of graph's paths in order to avoid paths' conflicts and warrant the uniqueness of a given path compared to another.

Figure 8 illustrates an extract of data mapping, initially in XML-EAD [19] and that corresponds to the first theme of our semantic modelling (see Figure 4). In XML-EAD, for instance, dimensions of an object is obtained with xpath */ead/archdesc/did/physdesc/dimensions* or */ead/archdesc/dsc/c/did/physdesc/dimensions*, according to the target information level that can be either the collection level or the document level.

Hence, to describe the dimensions of an object, we use a sequence of triples of the form:

[http://www.modref.org/biblionum/document_id/e70_thing, rdf : type, ecrm : E70_Thing],

[http://www.modref.org/biblionum/document_id/e70_thing, ecrm : P43_has_dimension, http://www.modref.org/biblionum/document_id/e54_dimension],

[http://www.modref.org/biblionum/document_id/-e54_dimension, rdf : type, ecrm : E54_Dimension],

[http://www.modref.org/biblionum/document_id/-e54_dimension, rdfs : label, "/ead/archdesc/dsc/c/did/physdesc/dimensions"],

[http://www.modref.org/biblionum/document_id/e71_man-made_thing, owl : sameAs, http://www.modref.org/biblionum/document_id/e70_thing].

Finally, the mapping realised will be translated into a programmatic data structure that will be then used to automatically generate files that follow RDF and CIDOC-CRM syntax: it is the data migration into triplestores or the creation of triplestores.

V. Data Migration into Triplestores and Factorisation

Efficiently migrating data into triplestores involves various skills. The sustainability of the whole procedure implies to define a general and rigorous architecture for the workflow of the different types of data handled. This architecture explicits the global method applied to all projects that wish to move their data into triplestores. This method is subdivided in different well-identified steps: data preparating (study and structural description), semantic modelling and data mapping from structural to semantic description and at last creating and exhibiting triplestores that can then be visualised and queried by users or semantic web applications. Initially, data are often non-structured or semi-structured (notes, reports, books, html) and first need to be converted into a structured representation (spreadsheets, databases, XML files) in order to easily construct their semantic representation thereafter. This continuum of steps requires several skills and needs some of in-between profiles skills to enable moving data from one format to another: (1) non-structured data or semi-structured data to structured data; (2) structured data to semantic data.

Besides, the key element of the architecture for migrating data into triplestores is the modelling and mapping of data with the semantic graph model chosen. In order to achieve data migration of ModRef project, we have performed an alignment between their data structural description and their semantic description by filling some nodes of the semantic graph with data retrieved from databases or collections of XML-EAD files. This migration into triplestores implies at the same time the reading of databases and the parsing of XML-EAD files (see Table I). Nodes filled with values are terminal nodes and non-terminal nodes are filled with URIs.

On the other hand, on instances, when we are mapping data with the CIDOC-CRM model, we do not need to explicit all the hierarchy of the classes used since we already have an implementation of the whole model of the CIDOC-CRM ontology. Subclasses inherit properties from all their superclasses. Hence, we can perform what we call here a *factorisation* and then reduce de number of use of the identity property *owl:sameAS* between the hierarchy of classes.

A. Factorisation

For the first modelling theme *"General characteristics of objects"*, the simplification is straightforward. *E24 Physical Man-Made Thing* is the most specialized reference class. Then all its super-properties are inherited and we finally get Figure 9. The other main simplification is that we no longer need to keep track of the hierarchy from *E24 Physical Man-Made Thing* to *E1 CRM Entity*. Hence, this is going to reduce the number of non terminal nodes in our triplestores and we can guess easily that the number of terminal nodes will remain the same.

For the second theme *"Events of beginning of existence (origin) and end of existence"*, the simplification is two-fold because we have a persistent part (*E24 Physical Man-Made Thing*) and a temporal part (*E63 Beginning of Existence* or *E64 End of Existence*) to deal with at the same time. Then all super-properties of classes *E24 Physical Man-Made Thing*, *E63 Beginning of Existence* or *E64 End of Existence* are inherited and in the end we obtain Figure 10. The other main simplification is that we no longer need to keep track of the hierarchy of our three main classes *E24 Physical Man-Made Thing*, *E63 Beginning of Existence* or *E64 End of Existence*. Hence, as for the first modelling theme, this is going to reduce the number of non terminal nodes in our triplestores and we can also guess easily that the number of terminal nodes will remain the same.

For the third theme *"Modelling of miscellaneous activities"*, the simplification is the same as for the second theme because we have also a persistent part (*E24 Physical Man-Made Thing*) and a temporal part (miscellaneous activities with classes such as: *E10 Transfer of Custody, E13 Attribute Assignement, E16 Measurement, E87 Curation Activity*). Once again, the consequences are the same, all super-properties of target classes *E24 Physical Man-Made Thing* or *E10 Transfer of Custody* for instance are inherited and finally we obtain Figure 11, which illustrates only the persistent item *E24 Physical Man-Made Thing* and the activity *E10 Transfer of Custody* though we can easily guess that the model remain the same for others activities (*E13 Attribute Assignement, E16 Measurement, E87 Curation Activity*). Of course, the other main simplification is that we no longer need to keep track of the hierarchy of our target main classes *E24 Physical Man-Made Thing*, *E10 Transfer of Custody, E13 Attribute Assignement, E16 Measurement, E87 Curation Activity*. Hence, this is going to reduce the number of non terminal nodes in our triplestores and we can also guess easily that the number of terminal nodes will remain the same.

For the fourth theme *"Modelling of inscriptions and other representations"*, the simplification is two-fold between a persistent part (*E24 Physical Man-Made Thing*) and another persistent part (inscriptions and other representations represented

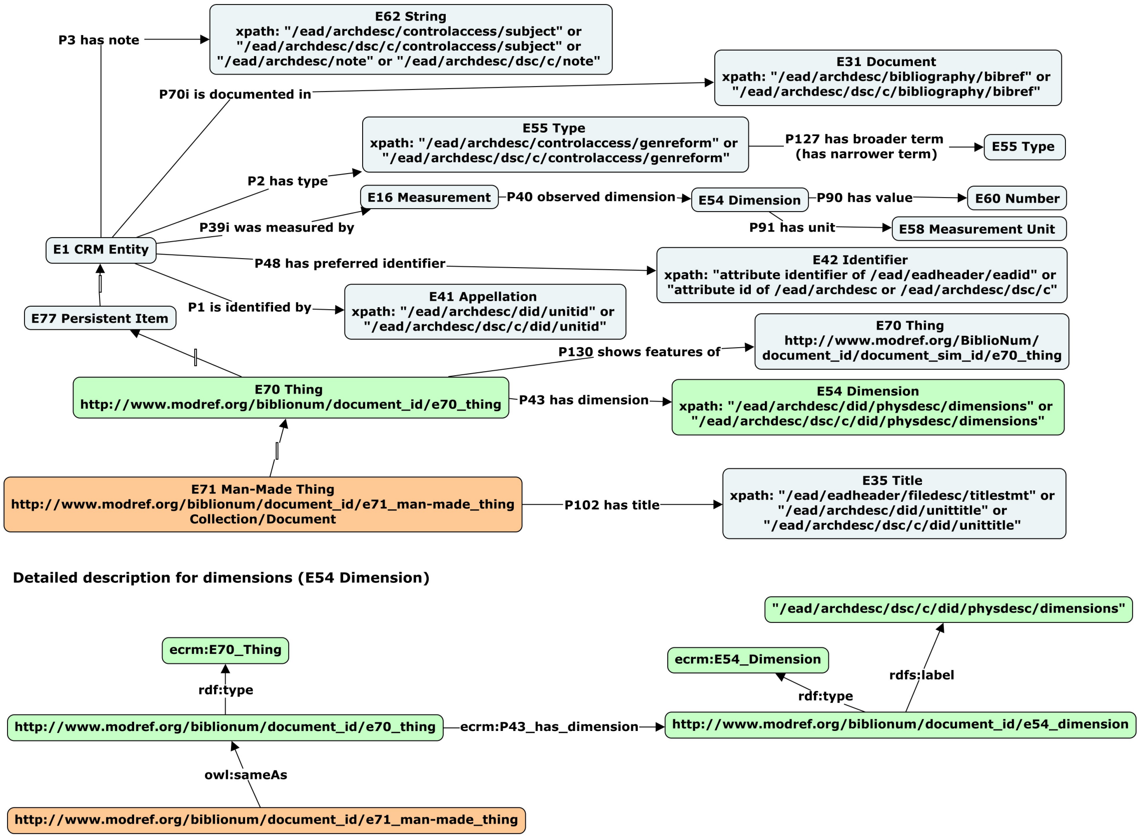

E62 String
xpath: "/ead/archdesc/controlaccess/subject" or "/ead/archdesc/dsc/c/controlaccess/subject" or "/ead/archdesc/note" or "/ead/archdesc/dsc/c/note"
P3 has note
E31 Document
xpath: "/ead/archdesc/bibliography/bibref" or "/ead/archdesc/dsc/c/bibliography/bibref"
P70i is documented in
E55 Type
xpath: "/ead/archdesc/controlaccess/genreform" or "/ead/archdesc/dsc/c/controlaccess/genreform"
P127 has broader term (has narrower term)
E55 Type
P2 has type
E16 Measurement
P40 observed dimension
E54 Dimension
P90 has value
E60 Number
P91 has unit
E58 Measurement Unit
P39i was measured by
E1 CRM Entity
P48 has preferred identifier
E42 Identifier
xpath: "attribute identifier of /ead/eadheader/eadid" or "attribute id of /ead/archdesc or /ead/archdesc/dsc/c"
E41 Appellation
xpath: "/ead/archdesc/did/unitid" or "/ead/archdesc/dsc/c/did/unitid"
P1 is identified by
E77 Persistent Item
E70 Thing
http://www.modref.org/BiblioNum/document_id/document_sim_id/e70_thing
P130 shows features of
E70 Thing
http://www.modref.org/biblionum/document_id/e70_thing
P43 has dimension
E54 Dimension
xpath: "/ead/archdesc/did/physdesc/dimensions" or "/ead/archdesc/dsc/c/did/physdesc/dimensions"
E71 Man-Made Thing
http://www.modref.org/biblionum/document_id/e71_man-made_thing
Collection/Document
P102 has title
E35 Title
xpath: "/ead/eadheader/filedesc/titlestmt" or "/ead/archdesc/did/unittitle" or "/ead/archdesc/dsc/c/did/unittitle"
Detailed description for dimensions (E54 Dimension)
"/ead/archdesc/dsc/c/did/physdesc/dimensions"
ecrm:E70_Thing
ecrm:E54_Dimension
rdfs:label
rdf:type
rdf:type
http://www.modref.org/biblionum/document_id/e70_thing
ecrm:P43_has_dimension
http://www.modref.org/biblionum/document_id/e54_dimension
owl:sameAs
http://www.modref.org/biblionum/document_id/e71_man-made_thing

Figure 8. Data mapping snippet between XML-EAD and CIDOC-CRM.

by the super-class *E90 Symbolic Object*). Indeed, there is an interaction between the two persistent classes *E24 Physical Man-Made Thing* and *E90 Symbolic Object* thanks to the property *P128 carries*. In this specific case of modelling inscriptions and other representations of a given object, *inscriptions (E36 Visual Item or E33 Linguistic Object, E34 Inscription)* are sub-classes of class *E90 Symbolic Object* (see Figure 12). If we are interested in any of *E90 Symbolic Object* sub-classes then we can move the property *P128 carries* form class *E24 Physical Man-Made Thing* to our target specific class (rule of inheritance). Hence, this will reduce that specific part of the hierarchy and also the number of non terminal nodes in this specific part. On the other hand, *other representations* of a given object is considered and described the same way as the original or target object using interactions through properties such as: *P62 depicts, P65 shows visual item, P138 represents* (see Figure 12).

The proof of concept of ModRef or the validation of data migration into triplestores is a set of tasks uphill (preparating and structuring of data, semantic modelling, alignment of data) and downhill (publishing, visualising, querying and exploring triplestores) the migration process. Hence, exploiting triplestores by querying and exploring them and the avantages that can be then got through triplestores is the other major aspect around those new data warehouses of RDF documents.

VI. Visualisation and Exploitation of triplestores

Created triplestores are available for visualising (under three different ways: rdf, triples and attribute-value summary) and querying through our web application. The interest of triplestores is that we have a public and published model of information representation that enables querying triplestores indifferently with the same procedures. We have defined two procedures for exploiting triplestores: interfaces similar to ”general query forms” and ”Endpoint Sparql” (see Figure 13).

As they are really close to natural language, ”general query forms” are simple and intuitive means for querying triplestores. Special knowledge is not necessary, all that is needed is to fill target fields on a given form and launch the query execution. A Sparql query is automatically constructed using values of filled fields and the query obtained is then used to retrieve information from triplestores. At the end of query execution, a list of objects is selected and returned as results to the user who can then visualise them in three ways: *rdf, triples and attribute-value summary*.

Besides, we can also query triplestores by using ”Endpoint Sparql”. This second query mode requires Sparql query language knowledge that is, at present, the reference language for querying RDF documents. Sparql is a simple language but not always at everyone’s comprehension level. Hence, ”general query forms” can be seen as a first query step for triplestores whereas ”Endpoint Sparql” guarantee a more complete exploitation of triplestores by a free formulation of

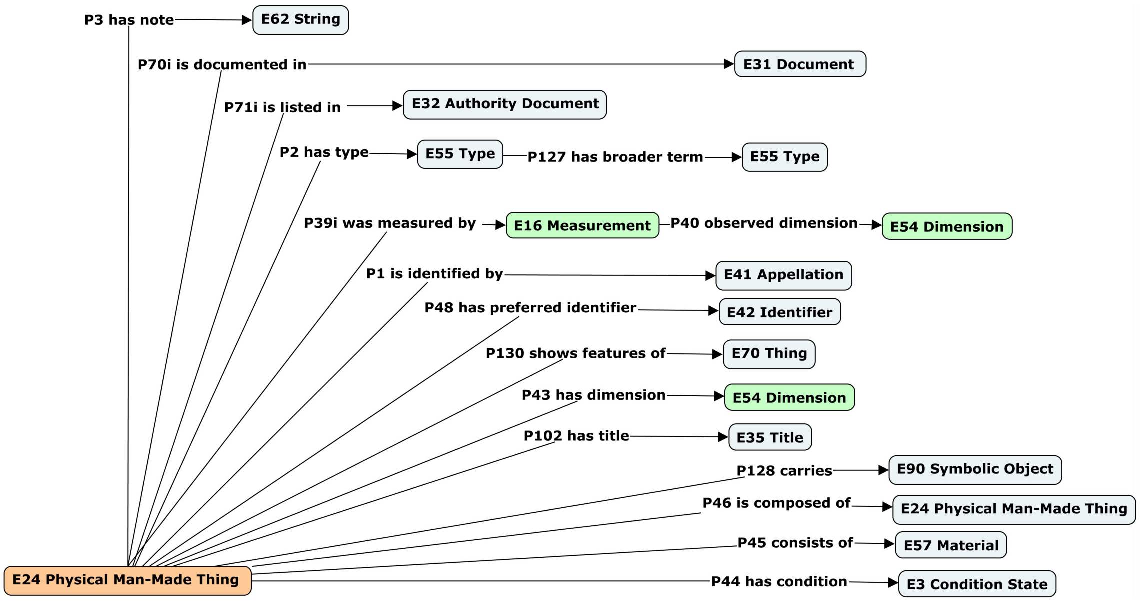


Figure 9. Factorisation of the modelling of general characteristics.

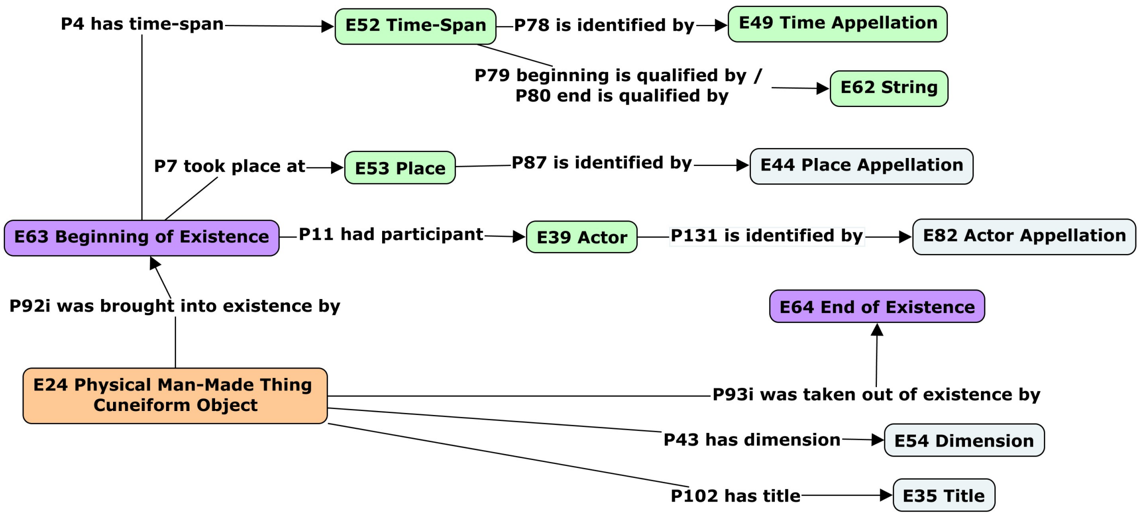


Figure 10. Factorisation of modelling of events of beginning and end of existence.

Sparql "Select" query type.

Our web application allows to visualise, query and explore triplestores separately for each pilot project of ModRef but also together by using the LOD (Linked Open Data) of ModRef. The web application provides, for each project and for the LOD of ModRef, a service to visualise triplestores data under three forms but also provides a service to query triplestores by using either "general query forms" or "Endpoint Sparql". Hence, as results to a query, the LOD allows to retrieve various information (statue, tablet) coming from different triplestores (see Figure 13). Several Sparql queries have been executed to validate data migration and a list of queries samples is provided with the web application. We have developped our own web application "Endpoint Sparql" and we also provide a Virtuoso "Endpoint Sparql" (Virtuoso is a software that allows to create an instance of "Endpoint Sparql") [25].

Note that the notion of exploiting triplestores refers to notions of querying and exploring semantic graphs. Hence, querying triplestores is executing Sparql queries that are pre-defined (general query forms) or free (Endpoint Sparql) whereas exploring triplestores is a kind of querying only performed through "Endpoint Sparql" for it allows to discover various paths in a semantic graph towards a given data. Actually, different paths can lead to the same information inside a graph (by the use of various notions: shortcut, refinement, inheritance/entailment, inverse) even if those paths are not always all filled. We can then write Sparql queries to discover if different paths that lead to a given data exist or write queries to know paths that lead to terminal nodes associated to values. Exploring is then very important to master a specific

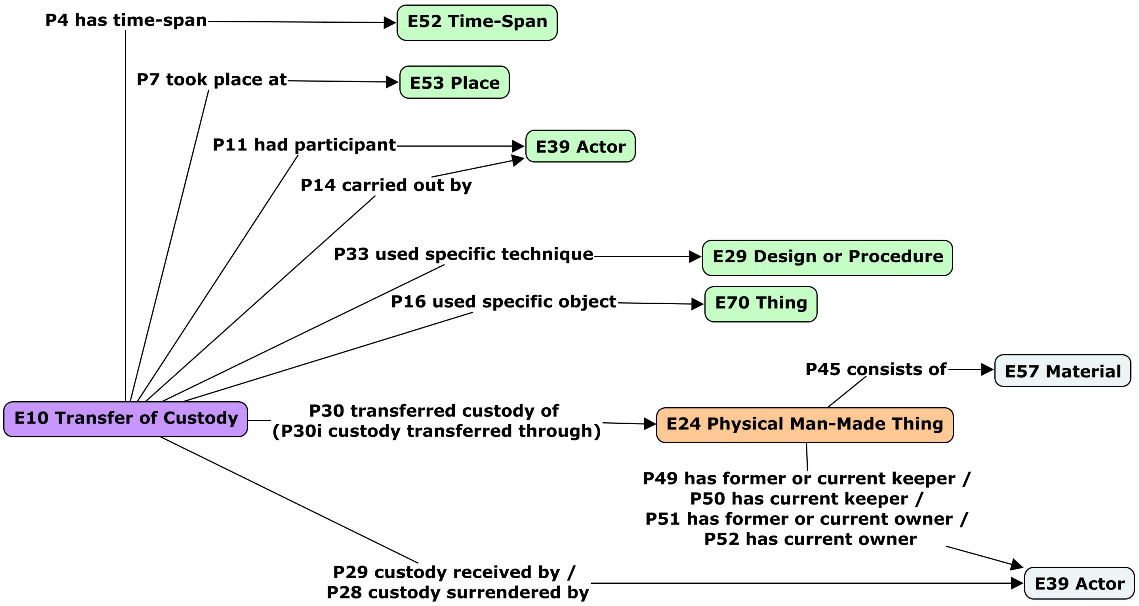


Figure 11. Factorisation of modelling of miscellaneous activities (eg. Transfer of Custody).

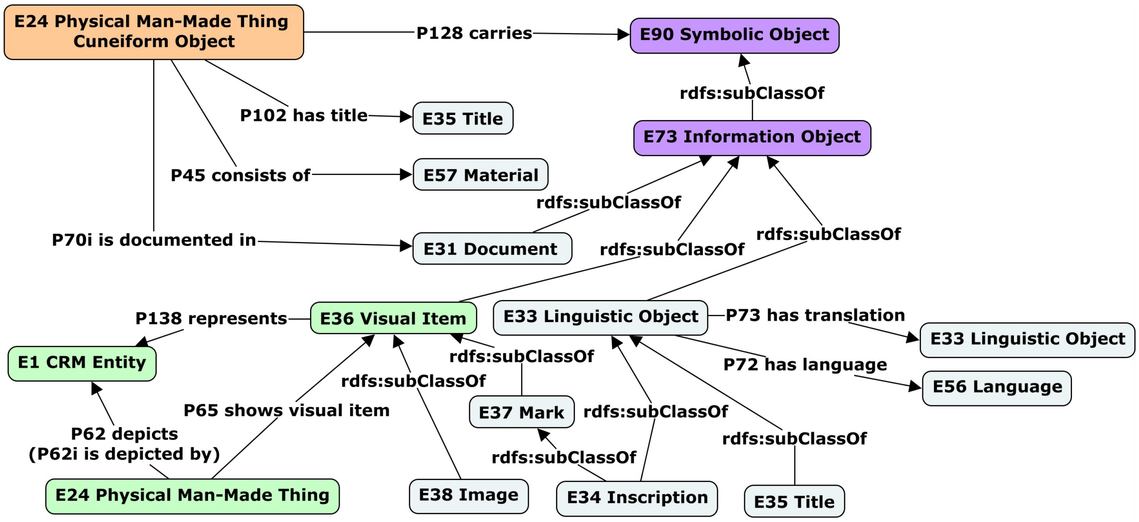


Figure 12. Factorisation of modelling of inscriptions and other representations of an object.

triplestore.

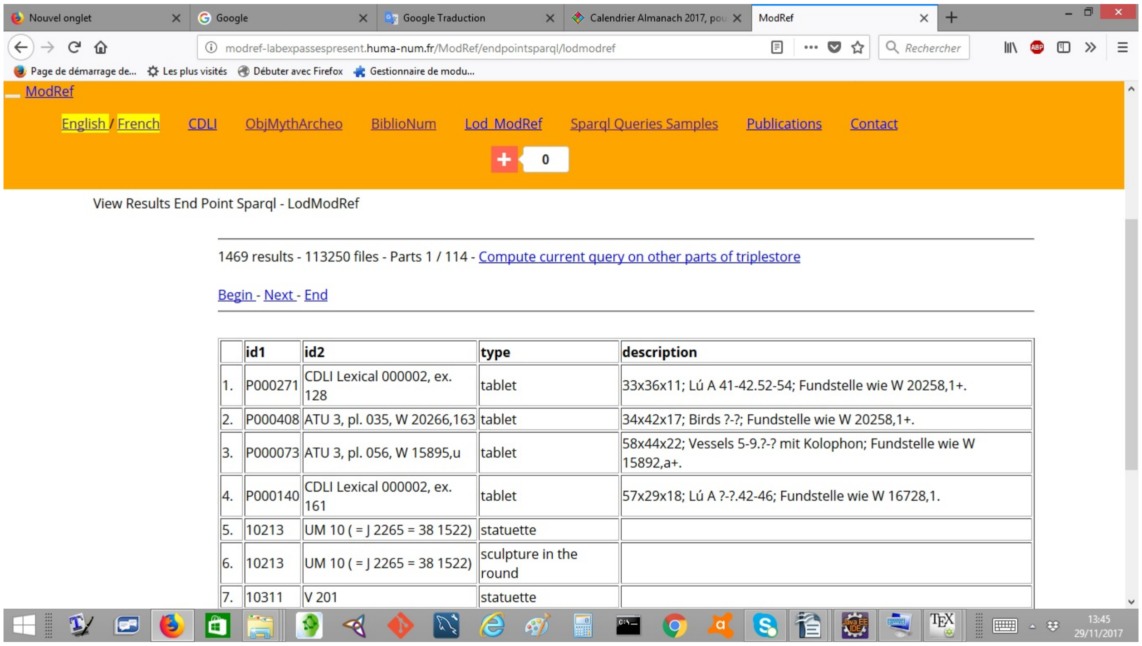
VII. Evaluation procedure and results

We have perform several Sparql queries types to validate the various datasets of our triplestores. The queries can be divided into two groups, one group related to the general RDF syntax schema (*list of concepts or predicates used, list of terminal triples (see Table II), list of triples of a given resource, extracts of paths leading to non-empty terminal nodes*) and another group related to the CIDOC-CRM schema (*checking instanciation of a given class, checking labels of given entity or resource (see Table III), characteristics of a given object (see Table IV), information on origin or custody of a given object*).

TABLE II. LIST OF TERMINAL TRIPLES WHERE THE TERMINAL NODE IS NON-EMPTY.

```
SELECT Distinct ?subject ?predicate ?object
WHERE
{
?subject ?predicate ?object .
Filter ( isLiteral(?object) && ?object != "" )
}
```

Moreover, triplestores are subdivided into constant parts (number of objects or triples) and queries are executed each time on one part and gradually on the other parts if the user asks so. The results are then progressively merge. The user chooses to execute its queries bit by bit and can stop the execution on any part of the triplestore. The current part



View Results End Point Sparql - LodModRef

1469 results - 113250 files - Parts 1 / 114 - Compute current query on other parts of triplestore

Begin - Next - End

	id1	id2	type	description
1.	P000271	CDLI Lexical 000002, ex. 128	tablet	33x36x11; Lú A 41-42.52-54; Fundstelle wie W 20258,1+.
2.	P000408	ATU 3, pl. 035, W 20266,163	tablet	34x42x17; Birds ?-?; Fundstelle wie W 20258,1+.
3.	P000073	ATU 3, pl. 056, W 15895,u	tablet	58x44x22; Vessels 5-9.?-? mit Kolophon; Fundstelle wie W 15892,a+.
4.	P000140	CDLI Lexical 000002, ex. 161	tablet	57x29x18; Lú A ?-?.42-46; Fundstelle wie W 16728,1.
5.	10213	UM 10 (= J 2265 = 38 1522)	statuette	
6.	10213	UM 10 (= J 2265 = 38 1522)	sculpture in the round	
7.	10311	V 201	statuette	

Figure 13. ModRef Project Web Application: Endpoint Sparql.

TABLE III. LIST OF TYPES OR CATEGORIES OF OBJECTS.

```
PREFIX ecrm: <http://erlangen-crm.org/150929/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT Distinct ?type
WHERE
{
?type_uri rdf:type ecrm:E55_Type .
?type_uri rdfs:label ?type .
Filter ( ?type != ”” )
}
```

TABLE IV. LIST OF CHARACTERISTICS THAT COME FROM ROOT ENTITY ”E1 CRM Entity”.

```
PREFIX ecrm: <http://erlangen-crm.org/150929/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT Distinct ?id1 ?id2 ?type ?description
WHERE
{
?e1_obj ecrm:P48_has_preferred_identifier ?id1_uri .
?id1_uri rdfs:label ?id1 .
?e1_obj ecrm:P1_is_identified_by ?id2_uri .
?id2_uri rdfs:label ?id2 .
?e1_obj ecrm:P2_has_type ?type_uri .
?type_uri rdfs:label ?type .
?e1_obj ecrm:P3_has_note ?description .
}
```

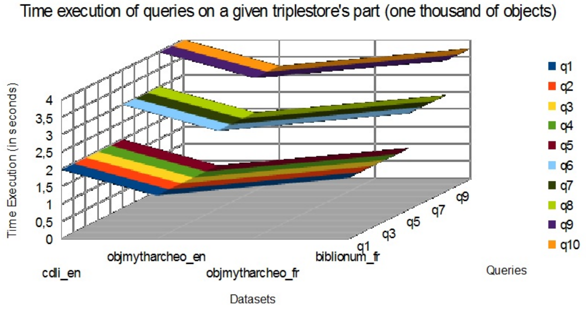


Figure 14. Queries Execution for ModRef Project.

number (on which the current query has been executed) and the total number of parts are always shown. Figure 14 shows that queries average time execution (in seconds) is rather constant on a given triplestore's part (*1000 objects*) and the execution speed of these queries is quite good for the users. Therefore, the cumulative time execution increases as triplestore's parts are combined.

TABLE V. MIGRATION PROCEDURE GENERAL STATISTICS.

	CDLI	ObjMythArcheo	BiblioNum
Number of Logical Structure Queries	1 SQL query	32 SQL queries	36 XML-EAD paths
Number of Literal Values	5 300 000	280 000	930 000
Number of Concepts	36	36	36
Number of Predicates	39	39	39

Once the modelling and alignment tasks were done, we proceeded to the implementation of the architecture of the migration process and thereafter we implemented the modules for visualising and querying (General Query Form, Endpoint Sparql) triplestores. All these implementations took about one year. The time execution of the migration process itself, on a 2.13 GHz processor with 4 GB RAM, takes: 6 hours for the project *CDLI* involving one long SQL query; 30 minutes for the project *ObjMythArcheo* involving 32 SQL queries; 2 hours for the project *BiblioNum* involving 36 XML-EAD paths.

Table V describes general statistics of the migration results for the three pilot projects of ModRef based on the following criteria: number of logical structure queries, number of literal values, number of concepts and predicates. Finally, the factorisation procedure has reduced the number of non-terminal nodes by around 30 percent.

VIII. Conclusion

The ModRef project allows to realise the proof of concept (POC) of data migration into CIDOC-CRM triplestores through: a general architecture that identifies the various steps; modelling and data mapping with the CIDOC-CRM semantic graph; data migration into triplestores; publishing of triplestores through a bilingual ''*English-French*'' web application [8] that provides services for visualising, querying and exploring triplestores.

Further work is directed towards:

1) *integration to others existing internet LOD (Linked Open data)* [26][27] in order to improve the sharing, exchange and discovery of knowledge at a greater scale. LOD should increase the discovery of new knowledge, because of the amount and diversity of linked data but mainly due to the use of semantic web formalisms, metadata languages, thesaurus published, standardised and even normalised;
2) *comparison of various triplestores (DBPedia, Nakala, British Museum, ModRef)* that describe similar data [28] (similar objects, objects of same historical period, objects of same type, identical objects) in a LOD context. This will lead to mutual enrichment of the various actors of the LOD;
3) *inference implications or issues or consequences* on CIDOC-CRM triplestores as well as on integration with others triplestores using factorised or non-factorised CIDOC-CRM triplestores.

Acknowledgment

The author would like to thank the laboratory Labex ''Les passés dans le présent'' of the University of Paris 10 Nanterre and the ANR project ModRef referenced by ANR-11-LABX-0026-01.

References

[1] P. L. Tchienehom, "Modref project: from creation to exploitation of cidoc-crm triplestores," The Fifth International Conference on Building and Exploring Web Based Environments (WEB 2017), 2017, pp. 52–60.

[2] D. Oldman, M. Doerr, G. de Jong, B. Norton, and T. Wikman, "Realizing lessons of the last 20 years : A manifesto for data provisioning and aggregation services for the digital humanities. http://www.dlib.org/dlib/july14/oldman/07oldman.html [retrieved: November, 2017]," D-Lib Magazine, vol. 20, no. 7/8, 2014.

[3] P. L. Boeuf, M. Doerr, C. E. Ore, and S. Stead, "Definition of the cidoc conceptual reference model, version 6.2," Produced by the ICOM/CIDOC Documentation Standards Group, Continued by the CIDOC CRM Special Interest Group. http://www.cidoc-crm.org/ [retrieved: November, 2017], May 2015.

[4] S. V. Hooland and R. Verborgh, Linked Data for Libraries, Archives and Museums. How to Clean, Link and Publish Your Metadata, A. L. Association, Ed., 2014.

[5] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," IEEE Intelligent Systems, vol. 21, no. 3, 2006, pp. 96–101.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," Scientific American, 2001, pp. 34–43.

[7] T. Berners-Lee, "Axioms, architecture and aspirations," W3C all-working group plenary Meeting. https://www.w3.org/2001/Talks/0228-tbl/slide1-0.html [retrieved: November, 2017], 2001.

[8] "Modref cidoc-crm project," http://modref-labexpassespresent.huma-num.fr [retrieved: November, 2017] or http://triplestore.modyco.fr [retrieved: November, 2017].

[9] P. L. Tchienehom, Modélisation et Exploitation de Profils : Accès sémantique à des Ressources, E. U. Europennes, Ed., 2015, iSBN 978-3-8416-7617-7.

[10] R. Heartfield and G. Loukas, "A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks," ACM Computing Surveys (CSUR) - DL (Digital Library), vol. 48, no. 3, 2016.

[11] J. Scarinci and T. Myers, "A semantic web framework to enable sustainable lodging best management practices in the usa," Information Technology and Tourism, vol. 14, no. 4, 2014, pp. 291–315.

[12] "Cidoc-crm implementation," http://www.erlangen-crm.org/ [retrieved: November, 2017], 2015.

[13] "British museum cidoc-crm project," http://collection.britishmuseum.org/ [retrieved: November, 2017].

[14] "Arches cidoc-crm project," http://www.getty.edu/conservation/our_projects/field_projects/arches/ [retrieved: November, 2017].

[15] "Dbpedia project," http://www.dbpedia.org/sparql [retrieved: November, 2017].

[16] T. Ruan, Y. Li, H. Wang, and L. Zhao, "From queriability to informativity, assessing ''quality in use'' of dbpedia and yago," In proceedings of the 13th Extended Semantic Web Conference ESWC'16, 2016, pp. 52–68.

[17] "Nakala project," http://www.nakala.fr/sparql [retrieved: November, 2017].

[18] P. Haase, J. Broekstra, A. Eberhart, and R. Volz, "Comparison of rdf query languages," In proceedings of the third International Semantic Web Conference ISWC'04, 2004, pp. 502–517.

[19] "Xml-ead elements," https://www.loc.gov/ead/tglib/element_index.html [retrieved: November, 2017].

[20] "Cdli cidoc-crm project," http://www.cdli.ucla.edu [retrieved: November, 2017].

[21] "Objmytharcheo cidoc-crm project," http://www.limc-france.fr and http://medaillesetantiques.bnf.fr [retrieved: November, 2017].

[22] "Biblionum cidoc-crm project," http://www.argonnaute-u.paris10.fr [retrieved: November, 2017].

[23] D. Faria, C. Martins, and A. Nanavaty, "Agreementmakerlight results for oaei 2014," ISWC Workshop on Ontology Matching - Proceedings of the 9th International Conference on Ontology Matching (OM'14), vol. 1317, 2014, pp. 105–112.

[24] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. Cruz, and F. Couto, "The agreementmakerlight ontology matching system," The 12th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), 2013, pp. 527–541.

[25] "Modref virtuoso endpoint sparql," http://3s-passespresent.huma-num.fr/sparql [retrieved: November, 2017].

[26] W. Beek, L. Rietveld, S. Schlobach, and F. van Harmelen, "Lod laundromat: Why the semantic web needs centralization (even if we don't like it)," IEEE Internet Computing, vol. 20, no. 2, 2016, pp. 78–81.

[27] E. Daga, M. d'Aquin, A. Adamou, and S. Brown, "The open university linked data - data.open.ac.uk. semantic web," Semantic Web, vol. 7, no. 2, 2016, pp. 183–191.

[28] W. Beek, S. Schlobach, and F. van Harmelen, "A contextualised semantics of owl:sameas," In proceedings of the 13th Extended Semantic Web Conference ESWC'16, 2016, pp. 405–419.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org