Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 158–167

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A Tool to Explore the Population of a CIDOC-CRM Ontology

Davide Varagnolo[a,*], Dora Melo[b,c], Irene Pimenta Rodrigues[a,c]

[a]*Department of Informatics, University of Évora, Portugal*
[b]*Coimbra Business School—ISCAC, Polytechnic Institute of Coimbra, Portugal*
[c]*NOVA Laboratory for Computer Science and Informatics, NOVA LINCS , Portugal*

Abstract

This paper presents a visualising tool to explore the population of an Ontology, obtained through the processes of automatic migration and text information extraction. It was developed in the context of EPISA project, a R&D project that aims to represent the Portuguese National Archives records information in CIDOC-CRM, an ontology developed for museums. The tool allows the migration process developers to visualise the instances and their properties, and to debug the migration process and the migration representation model, or to explore the Archives by final users. It uses modeling and reasoners OWL-API with SPARQL-DL queries to obtain the exploration results.

Keywords: Archives; CIDOC-CRM; Ontology Visualization; Knowledge Representation; Semantic Web

1. Introduction

The present work was done in the context of project EPISA (Entity and Property Inference for Semantic Archives), a research project involving the Portuguese National Archives - Torre do Tombo (ANTT), a research project involving archival experts, and Information and Computer Science researchers.

The EPISA project aims to design a prototype, as an open-source knowledge platform, to represent archival information on a linked data model. One of the project's major tasks is the semantic migration, i.e, the process to extract and represent the relevant entities and their properties from the existing records in the actual DigitArq [25], the archive national system that uses well-established description standards, namely the ISAD(G) (General International Standard Archival Description) [9] and ISAAR(CPF) (International Standard Archival Authority Record for Corporate Bodies, Persons and Familie) [28] with a hierarchical structure adapted to the nature of archival assets. For this purpose, an automatic semantic migration prototype, based on Knowledge Discovery, from Digital Archive metadata to populate an ontology in CIDOC-CRM was developed. The data model and description vocabularies adopted are built upon

* Corresponding author.
E-mail address: dmelo@iscac.pt

1877-0509
10.1016/j.procs.2021.08.017

the CIDOC-CRM (Conceptual Reference Model) standard, an ontology developed for museums by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) [20, 8].

Ontology Population consists of the process of updating an ontology with instances and relations between them from an input structured or unstructured knowledge resource. The instances and their relations are represented in a structured format, reflecting the information and context of the knowledge resource, and therefore they became part of the existing knowledge in the ontology [24, 15, 10, 12].

The automatic semantic migration strategy adopted is built on a set of Mapping Description Rules to map the information between DigitArq and CIDOC-CRM. This strategy also includes the representation of information extracted from text. The Mapping Description Rules, in both tasks, are defined manually and, for the first task, take into account the ISAD(G) description specificities of the information presented in DigitArq. The formalism of the Mapping Description Rules is well-structured in the way that it is possible to adapt to other sources or destinations and can be automatically interpreted. This formalism allows to easily adapt the migration of the DigitArq information to other archives, enabling the integration of all the Portuguese Archives through CIDOC-CRM representation, and also integrate with other CIDOC-CRM representation domains, such as Museums, Archaeology, Architecture, and Cultural Heritage. Regarding the information extraction from text, it is not intended to extract all the information, but only parts of information considered important, such as baptisms, births, inventories due to death, transfers of documents between archives, institutions, persons, and places involved in those events.

Methodologies to extracted general information from text into ontologies are presented in several works, such as [16, 4, 18]. In particular, OntoPrima is a NLP-based Ontology Population system that extracts instances of concepts and relations from text to populate an ontology using NLP techniques.

The goal of this paper is to present a visualising tool to explore the Population of an ontology, obtained through the processes of automatic migration or text information extraction. The tool allows the developers of the migration process to visualise the instances, their individual object properties with other instances and also their data properties, taking into account some specificities of the CIDOC-CRM ontology, such as the use of the class 'E_{55} Type'. The tool also can be used to debug the migration process and the representation model implicit in the Mapping Description Rules, or to explore the Archives by final users. Modeling and reasoners OWL-API with SPARQL-DL queries are used to obtain the exploration results.

The process to automatically populate the CIDOC-CRM ontology is based on Mapping Description Rules to semantically translate the archives descriptive information into CIDOC-CRM representation [21]. These rules express how the ontology individuals are related to each other and how it is possible to query the ontology population in order to retrieve information. Therefore, the visualisation tool presented in this paper is supported by a set of queries based on the Mapping Description Rules set to allow explore the Population of the ontology.

The development of interfaces as a tool to access, retrieve or manipulate OWL (the W3C Web Ontology Language (https://www.w3.org/TR/owl-features/)) ontologies are not new. Over the last decades, graphical interfaces have evolved in an increasingly friendly way. However, the interfaces available for Semantic Web are focused in the mainly aspects, such as the ontology design, like concepts and properties defined, which turns very difficult the population exploration itself for an ordinary user. Examples of such interfaces are OntoGraf (https://protegewiki.stanford.edu/wiki/OntoGraf), a Protégé plugin that gives support for interactively navigating the relationships of OWL ontologies; or VOWL (http://vowl.visualdataweb.org/protegevowl.html) [13], also a Protégé plugin for the user-oriented visualization of ontologies, and implements the Visual Notation for OWL Ontologies (VOWL) by providing graphical depictions for elements of the OWL that are combined to a force-directed graph layout representing the ontology.

More recent and regarding CIDOC-CRM knowledge bases, some interfaces were developed, mainly in the cultural heritage domain, such as OpenArcheo [17], that allows the users to create complex query with a user's friendly GUI and facilitates the task of searching for information that users seek to find, or even Archives heritage inventory and management system [23]; and ONTOME a collaborative ontology management environment [2, 1]. An example of a distinct tool is the interface for manipulating narratives, Narrative Building and Visualisation Tool (https://dlnarratives.eu/tool.html) with the purpose of validating the Narrative Ontology [19], a conceptualisation of the domain of narratives and its specification expressed in first-order logic. This ontology has been implemented as an extension of the CIDOC-CRM, FRBRoo and OWL Time. It uses the SWRL rule language to express the axioms. The validation of the ontology has been performed in the context of the Mingei European project, in which it is applied

Table 1. Mapping Description Language Rules

Rule	Left part (rec[attribute value list])	Right part CIDOC-CMR
1	DigitArq(Rec)	$E_{31}\{= ID_{E_{31}}\} \rightarrow P_{70} \rightarrow E_{22}\{= ID_{E_{22}}\} \rightarrow P_{67} \rightarrow E_{33}\{= ID_{E_{33}}\}$
2	['Description level', V]	$\$ID_{E_{31}} \rightarrow P_2 \rightarrow (\langle E_{55}\{= V\}\rangle \rightarrow \langle P_2\rangle \rightarrow \langle E_{55}\{\text{= 'Description level'}\}\rangle)$
3	['Reference code', V]	$\$ID_{E_{31}} \rightarrow P_1 \rightarrow (E_{42}\{= V\} \rightarrow P_2 \rightarrow \langle E_{55}\{\text{= 'Reference code'}\}\rangle)$
4	['Scope and content',V]	$\$ID_{E_{31}} \rightarrow P_{01i} \rightarrow (PC_3 \rightarrow P_{02} \rightarrow \langle E_{62}\{= V\}\rangle \rightarrow P_{3.1} \rightarrow \langle E_{55}\{\text{= 'Scope and content'}\}\rangle)$
5	['Recipient', V]'	$\$ID_{E_{31}} \rightarrow P_{01i} \rightarrow (PC_{129} \rightarrow P_{02} \rightarrow (E_{21} \rightarrow P_1 \rightarrow \langle E_{41}\{= V\}\rangle) \rightarrow P_{129.1} \rightarrow \langle E_{55}\{\text{= 'Recipient'}\}\rangle)$
6	['Title and Type',(T_i, T_y)]	$\$ID_{E_{31}} \rightarrow P_{01i} \rightarrow (PC_{102} \rightarrow P_{02} \rightarrow \langle E_{35}\{= T_i\}\rangle) \rightarrow P_{102.1} \rightarrow \langle E_{55}\{= T_y\}\rangle)$
7	['Creation date', V]	$\$ID_{E_{31}} \rightarrow \langle P_{94}\rangle \rightarrow (E_{65} \rightarrow P_4 \rightarrow (E_{52} \rightarrow P_2 \rightarrow \langle E_{55}\{\text{= 'Creation date'}\}\rangle \rightarrow P_{170i} \rightarrow \langle E_{61}\{= V\}\rangle))$
8	['Modification date',V]	$\$ID_{E_{31}} \rightarrow \langle P_{94}\rangle \rightarrow (E_{65} \rightarrow P_4 \rightarrow (E_{52} \rightarrow P_2 \rightarrow \langle E_{55}\{\text{= 'Modification date'}\}\rangle) \rightarrow P_{170i} \rightarrow \langle E_{61}\{= V\}\rangle))$
9	['Hierarchy',(Root$_{ref1}$,Son$_{ref2}$)]	$\|\|E_{31}\|\| \rightarrow \|\|P_1\|\| \rightarrow \|\|E_{42}\{= \texttt{Root}_{ref}\}\|\| \rightarrow P_{106} \rightarrow (\|\|E_{31}\|\| \rightarrow \|\|P_1\|\| \rightarrow (\|\|E_{42}\{= \texttt{Son}_{ref}\}\|\|)$
10	['Baptism', birth(Mother, Father, DBirth)]	$ID_{E_{31}} \rightarrow P_{67} \rightarrow (E_{67} \rightarrow P_{98} \rightarrow (\|\|E_{21}\|\| \rightarrow \|\|P_{02i}\|\| \rightarrow (\|\|PC_{129}\|\| \rightarrow \|\|P_{129.1}\|\| \rightarrow \|\|E_{55}\{\texttt{'Recipient'}\}\|\| \rightarrow \|\|P_{01}\|\| \rightarrow \$ID_{E_{31}})) \rightarrow P_{96} \rightarrow (E_{21} \rightarrow P_1 \rightarrow \langle E_{41}\{\texttt{Mother}\}\rangle) \rightarrow P_{97} \rightarrow (E_{21} \rightarrow P_1 \rightarrow \langle E_{41}\{\texttt{Father}\}\rangle) \rightarrow P_4 \rightarrow (E_{52} \rightarrow P_{170i} \rightarrow \langle E_{61}\{\texttt{DBirth}\}\rangle)$

to the representation of knowledge about Craft Heritage. The tool makes available a semi-automatic way to construct narratives, intended as semantic networks of events related to each other through semantic relations. Using SPARQL queries, it allows the user to visualise the knowledge in the graph in simple formats like tables, network graphs, and timelines. All these platforms are a mean to integrate different domain knowledge bases for interoperability.

The remainder of this paper is divided into the following sections. In Section 2, the approach to the automatic migration process from DigitArq records to CIDOC-CRM concepts, relations, and properties is presented. Section 3 describes in detail the approach to visualize the information represented as CIDOC-CRM instances of classes, object properties and data properties. Finally, in Section 4, conclusions, further work and a future evaluation are drawn.

2. Approach to the Automatic Population of CIDOC-CRM

In this section, the automatic migration process from Archives units of description metadata in DigitArq [25] to CIDOC-CRM concepts, relations, and properties is presented.

Each DigitArq representation of metadata archives units has a scheme complying ISAD(G) [9] and ISAAR [28] recommendations. The DigitArq information is organized according to a set of fields and their values. Among this set of fields, there are some that present atomic values, such as the "Reference code", the "Title", or the "Recipient", that do not require further interpretation. For these fields, the migration process can be done by applying a predefined set of rules establishing the mapping between ISAD(G) elements and CIDOC-CRM classes and properties. The mapping was defined using the Mapping Description Language Rules, presented in [3], with some extensions to enable the distinction between a query and an assertion in the interpretation of the rules.

Table 1 presents a subset of the rules used in the migration process. These rules are necessary in this process but they should also be used to recover the information in the new CIDOC-CRM knowledge base, to write the right query, in Description Logic (DL) or SPARQL, to obtain some information.

2.1. Mapping the ISAD(G) Elements into CIDOC-CRM

In this subsection, the rules of Table 1 are detailed to explain the CIDOC-CRM representation, that are used to represent the metadata of archives units. These representations follow the CIDOC-CRM recommendations, and similar approaches for representing archives and collections are presented in [3, 27, 14].

Rule 1 defines the CIDOC-CRM representation of an unit as a new 'E_{31} Document' class instance linked: to a new 'E_{22} Human-Made Object' class instance (representing the object that materializes the unit) by the object property

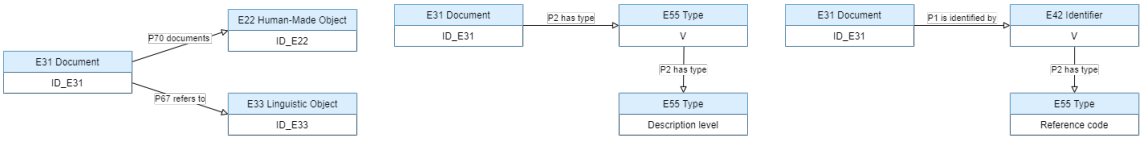

Fig. 1. (a) Unit of Description; (b) Description level; (b) Reference code.

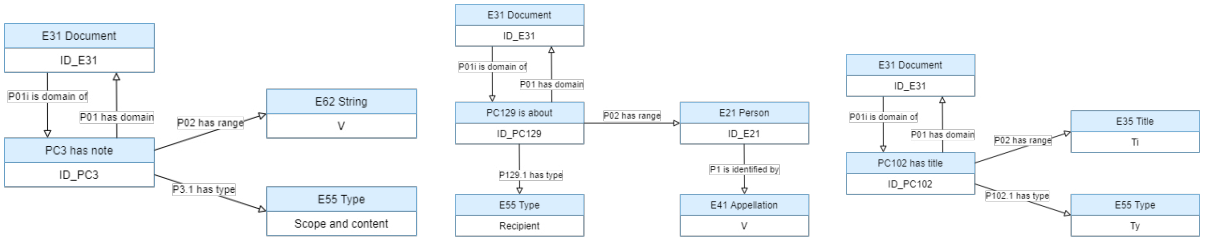

Fig. 2. (a) Scope and Content; (b) Recipient; (c) Title and its type.

'P_{70} documents'; and to a new instance of 'E_{33} Linguistic Object' (representing a conceptual object) by the object property 'P_{67} refers to'. Figure 1 *a)* has a visual representation of this rule.

Rule 2 associates the 'Description Level' to the instance that represents de unit ('E_{31} Document') through the data property 'P_2 has type'. The 'Description Level' is an ISAD(G) element whose value establishes the type of the unit of an archive, such as Fonds, Sub-Fonds, Series, Sub-Series, File, Item, etc. This value is represented has an instance of the class 'E_{55} Type', but only if the instance was not already created, the notion $\langle E_{55}\{= V\}\rangle$ represents it. Note that each 'Description Level' value will have the type 'Description Level'. This information about the type is not strictly necessary since, in the proposed representation, each 'E_{31} Document' will have one and only type. However, to explore the migrated information, it is important to enable SPARQL or DL queries about 'Description Level' values, as well as to use this type information as a tag in user interfaces, like the one proposed, or to interpret the term 'Description Level' in natural language queries interfaces. Figure 1 *b)* has a visual representation of this rule.

Rule 3 is used to represent the unit's 'Reference Code'. 'Reference Code' values are unique identifiers and a unit has only this one identifier. So, like with 'Description Level', the assertion on the type of the 'Reference Code' value is done with the same purpose of enabling direct queries for listing 'Reference Codes', inferring that a value is a reference code or interpreting the term 'Reference Code' in a natural language query. Figure 1 *c)* has a visual representation of this rule.

Rules 4, 5 and 6 are represented in Figure 2. These rules define the representation of the ISAD(G) elements: 'Scope and content', a text that is linked by a 'has note' property to the unit representation; 'Recipient', a person that is linked to the unit by an 'is about' property; and 'Title and Type', the unit title and the type of title that are linked to the unit by a 'has title' property. The issue with the representation of these elements is that a unit can have more then one property 'has note', 'is about' or 'has title', so the type of the relation must be represented in the relation. In OWL2, since relations must be binary, this has to be done by using a new entity. CIDOC-CMR already predicts these representations and these rules uses the CIDOC-CRM provided mechanism.

The representation of the ISAD(G) elements 'Creation Date' and 'Modification Date', see Figure 3 *a)* and *b*, is obtained by rules 7 and 8. The creation or modification date refers to the event of the creation of the unit metadata. So in these rules the event is created only if the 'E_{31} Document' that represents the unit does not have already a 'E_{65} Creation' linked, note the $\langle\rangle$ between the properties and classes in these rules.

Finally, rule 9 defines the representation of the hierarchic model of the archives, a fond is composed by a set of subfonds, files and/or series, a file is composed of a set of items, etc. In this rule the notations $\|P_i\|$ or $\|E_i\|$ mean that an instance of the property P_i or an instance of the class $\|E_i\|$ should exist already in the CIDOC-CRM Knowledge Base. Figure 3 *c)* presents the CIDOC-CRM representation of the hierarchy.

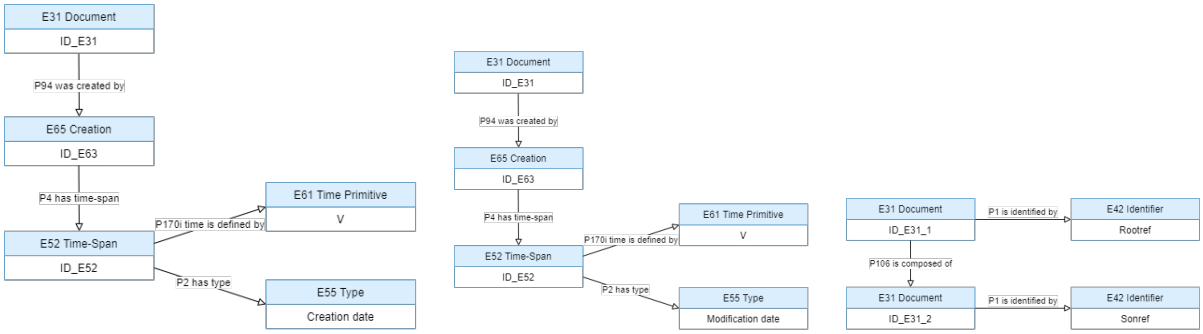

Fig. 3. (a) Creation Date. (b) Modification Date. (c) Hierarchy of Documents.

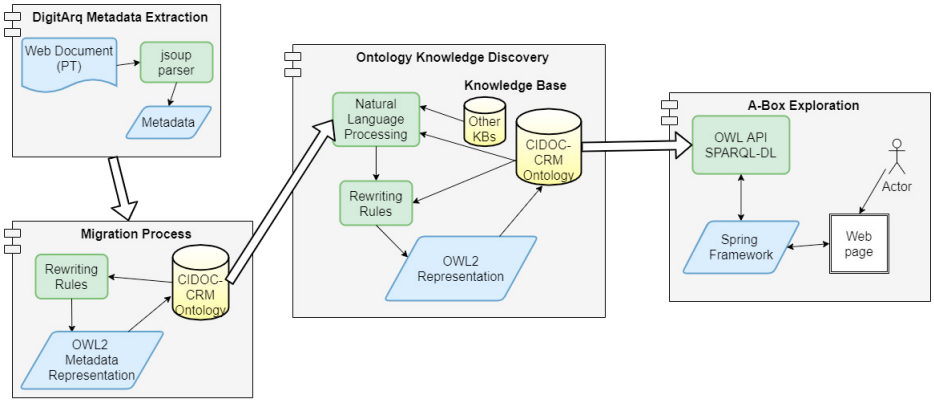

Fig. 4. Architecture of the Ontology Population and Visualization Tool.

2.2. *Architecture of the Ontology Population and Visualization Tool*

In this Subsection, a brief description of the architecture of the Ontology Population process and the Visualization Tool is presented. Exploring the Population of the ontology allows: on a first level, developers of the migration process to visualise the instances and their properties, as well as debug the migration process and the migration representation model; and on a second level, end-users to explore the Archive's collections. The OWL-Ontology Population is obtained through the processes of automatically migration or text information extraction of the Archives metadata into CIDOC-CRM ontology representation. The architecture of the full system is presented in Figure 4.

The A-Box Exploration process is supported by a query engine and serves as a middle layer application between the Graphical User Interface (GUI) and the Archival knowledge base. The question made by the user at the GUI level is translated to the corresponding CIDOC-CRM representation and the answer is retrieved using the query engine and then presented at the GUI application level, more details in Section 3.

The under layer of the system composes the complete migration process and it is done in three main steps: 1) DigitArq Metadata Extraction; 2) Migration Process; and 3) Ontology Knowledge Discovery. At first step, the metadata to be represented in CIDOC-CRM are extracted from the DigitArq database. Together with the database, a web-based search engine (web service) was developed to allow local and remote users to find and browse the Archive's collections. The resulting well-structured and normalised web service (`https://digitarq.arquivos.pt/`) shows, for each unit, the whole information needed to be considered in the migration process. For this purpose, the `jsoup` li-

brary[1] is used to extract web page content from specific fields. A process similar to the ones present in [6, 5], where jsoup library is also used to extract information from web pages to be analyzed and interpreted.

The second step represents the exact mapping process between the ISAD(G) elements and the CIDOC-CRM representations, and is made using the introduced Mapping Representation Rules (a summary presented in Table 1 and a dataset application example can be consulted in [22]). The OWL API [7] and the SPARQL-DL[26, 11] libraries are used to upload and model the CIDOC-CRM archival representation into a well-structured model for Java environment, to implement the mapping description rules, to update the mapping knowledge base, and also to reasoning over the knowledge base. The OWL API is a high level Application Programming Interface (API) for working with OWL ontologies, and is closely aligned with the OWL2 structural specification (https://www.w3.org/TR/owl2-syntax/). It supports parsing and rendering in the syntaxes defined in the W3C specification, manipulation of ontological structures, and the use of reasoning engines. The SPARQL-DL is a Java query engine, settled on top of the OWL API, and it is fully aligned with the OWL2 standard and adds a SPARQL-DL interface to every OWL API 3 reasoner.

Finally, the third step refers to the interpretation of some pieces of text provided by some ISAD(G) elements and that are not yet represented in the CIDOC-CRM ontology. This last step is done entirely over the information already represented in CIDOC-CRM, and obtained in the second step. The objective of the third step is to map valuable information to the knowledge base, by applying Natural Language Processing techniques to extract the additional information. and using a corresponding set of Mapping Representation Rules over the information extracted.

The interpretation and extraction of information from these texts depend on the type of information, such as baptisms, births, inventories due to death, transfers of documents between archives, institutions, persons, and places involved in those events. The ontology representation of the information and the corresponding mapping description rules are defined manually. An automatic text classifier and an automatic information extraction process of the information have been defined for each kind of information, which simplify the extraction process. The automatic text classifier is built using a manually annotated text from some ISAD(G) text elements as training dataset. The information extraction is done only on the text parts selected by the classifier, which typically have a natural language predefined structure and it is explored in the extraction process of entities and relations to populate the ontology. A Natural Language Pipeline is used in GATE(https://gate.ac.uk/) framework combined with Jape rules (https://gate.ac.uk/wiki/jape-repository/) to extracted the entities and relations. As an example, consider the text "*Pais: Manuel de Oliveira e Rufina Maria Data de nascimento: 10 de Fevereiro de 1812*" (Parents: Manuel de Oliveira and Rufina Maria Birthdate: 10th February, 1812"), retrieved from 'Scope and content' element of the "Record of the Baptism of Ana" (https://pesquisa.adporto.arquivos.pt/details?id=1374655), which refers to the baptism happening of the person named "Ana", 'Recipient' of the document. The result of the automatic text classifier applied to the referred piece of text is

['Baptism', birth("Rufina Maria", "Manuel de Oliveira", "10 de Fevereiro de 1812")]

and the corresponding Mapping Description Rule is Rule 10, Table 1.

2.3. Evaluation of the migration process

The migration process is composed of 3 modules (see Figure 4). The first two modules can be evaluated automatically and have to be correct, since the process consists of the transference of information from a consistent relational database into a semantic web representation.

The third module consists of representing the information contained in textual elements of ISAD(G), such as 'Scope and content'. This process is defined in 3 subprocesses: text classification, extraction, and representation. This task is not complete, since there is the need to identify more information to be extracted and define the rules to represent it in CIDOC-CRM. In the text classification subprocess, it is defined a classifier for each type of information to identify the texts where that kind of information could be extracted. These classifiers are built using machine learning and natural languages processing tools. For instance, to identify the texts that contain baptism information, a manually annotated sample of 200 Portuguese texts was considered. The classifier was built using a N-Gram TF-IDF model for the sample data, and a decision tree, allowing to obtain a high precision in the identification of the texts containing

[1] jsoup (http://jsoup.org) is a Java HTML Parser that provides a very reliable, user-friendly, and easy configuration and parameter adjustments capabilities, for connecting to URLs and extracting and manipulating data.

Fig. 5. Main Page of the Query Ontology Interface

information about baptisms. Regarding the extraction subprocess, in the classified text, the information is extracted into an established format (Rule 10, Table 1). This work is ongoing but some experiences were already performed, enabling the automatic extraction of some information. The Query Ontology Interface (Section 3) has an important role in the process of manual evaluation, enabling the visualization of the extracted information, and helps in identify the correctness information extracted and how it is related with the other ontology information.

3. Exploring the Population of the Archives Knowledge Base

The knowledge base exploration process is supported by an application program interface (API), entitled Query Ontology Interface, that facilitates the interaction between regular users and the knowledge base. The main reasons for the development of such API are to allow retrieving information from knowledge base without technically know how the information is represented in the ontology. The main target users, such as the librarians or archivists, are in general not able to make queries using SPARQL language, or even using description logic languages.

The Query Ontology Interface was developed using Spring Boot (https://spring.io/), a Java-based framework that allows to create a Graphical User Interface (GUI) and export the final API in a stand-alone application (originally a web-application). The SPARQL-DL Java query engine is used to search the knowledge base, and serves as a middle layer application between the GUI and the knowledge base, as shown in previous Section 2. The question made by the user at the GUI level is translated to the corresponding CIDOC-CRM representation and the answer is retrieved using the SPARQL-DL engine and then presented at the GUI application level. The use of a GUI to help users in the exploration process allowed to define the knowledge base query process as much user-friendly as possible.

The Query Ontology Interface, see Figure 5, main page offers the options:

Class. A menu with all the Classes of the ontology that have at least one instance in the knowledge base. The user can then select one of the classes to view all its instances, and also select one of these instances to view its content. The query beyond this menu is a general one and does not depend on the ontology representation.

```
SELECT DISTINCT ?class WHERE {Type(?ind,?class)}
```

Property. A list menu of all properties of the ontology that links two instances and allows the user to view the property instances selected. The query beyond this menu is a general one and does not depend on the ontology representation.

```
SELECT DISTINCT ?property WHERE {Property(?property),PropertyValue(?s,?property,?o)}
```

Reference Code. Search by a 'Reference Code' value and returns the document instance with the given identifier. This query search depends on the representation of the 'Reference code' and uses the information of Rule 3, Table 1.

```
SELECT ?doc WHERE {Type(?doc,<http://erlangen-crm.org/200717/E31_Document>),
    PropertyValue(?doc,<http://erlangencrm.org/200717/P1_is_i entified_by>,ReferenceCodeID)}
```

Keyword. Search by String on all the instances, as an example see Figure 6 (a). The result is a menu of instances where the String occurs, see Figure 6 (b) for the search of 'Ana'. Since SPARQL-DL has no defined operations axioms, the query to retrieve all the instances from the ontology is a general one, not depending on the ontology representation.

```
SELECT DISTINCT ?res WHERE {Individual(?res)}
```

Then, from the result set, it is extracted the instances where the String occurs, using a simple Java procedure.

Instance. The view of a selected instance. The result presents the instance value, its type and class, and the list of all object-properties, with its range, and object-inverse properties, with its domain, where the selected instance is domain or range, respectively. Except for the type, the search queries are general ones and do not depend on the ontology representation. The 'E_{55} type' in CIDOC-CRM is used to represent class information to avoid the creation of new

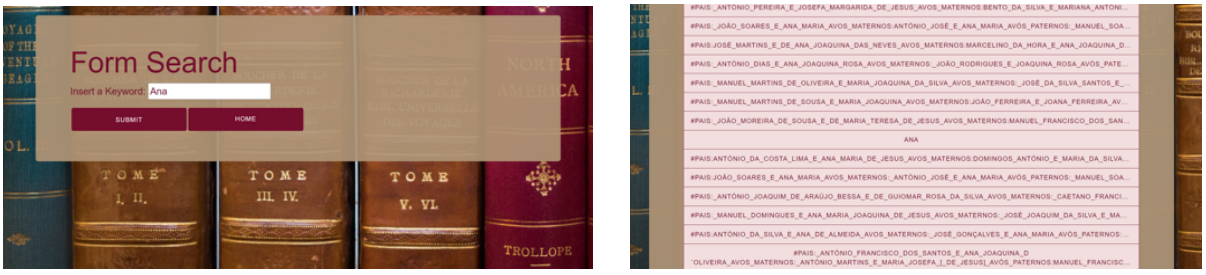

Fig. 6. (a) Search by String. (b) Result list of instances that match the String.

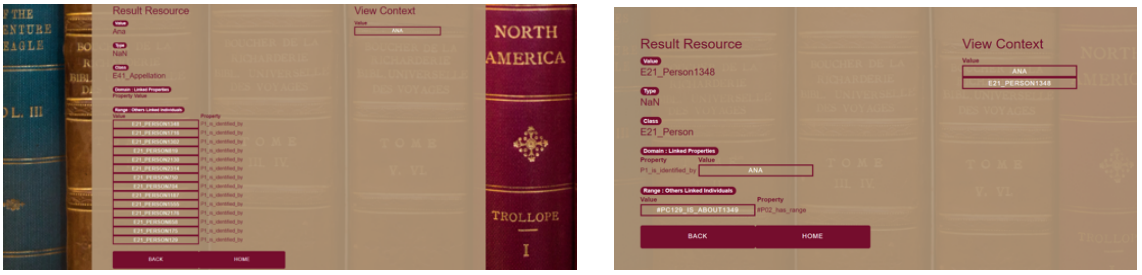

Fig. 7. (a) View of the instance 'Ana'. (b) View of the persons instances that are identified by 'Ana'.

Fig. 8. (a) View of the 'Recipient' of an instance of 'is about' (b) View of an instance of 'E_{31} Document'.

classes. Therefore, the type when is presented represents the instance class. Figure 8 (b) presents an instance of 'E_{31} Document' that is the type of 'Item'. Besides the information about the instance, the user also has the context view of its interaction, see Figure 7. The mapping rules enables to understand the information presented in the screen for an instance. In Figure 8 (b), the information of an instance of a 'Document' is presented. For instance and related with object properties: the first shows Rule 2; the second is the Rule 1; the third is the Rule 3; the fourth is Rule 1; and the fifth is the Rule 7. Regarding the inverse properties, the first is the Rule 5, the second is Rule 9, the third is the Rule 5, and the fourth is the Rule 6. Throughout the exploration of the individuals expressed by Figures 7 and 8, it is possible to infer that 'Ana' is the identification of a person which is the 'Recipient' of a document.

The query for obtaining an instance object property and its range is:

SELECT `?p ?val` **WHERE** `{ObjectProperty(?p), PropertyValue(InstanceURI,?p,?val)}`

The query for obtaining an instance object inverse property and its range is:

SELECT `?p ?val` **WHERE** `{ObjectProperty(?p), PropertyValue(?val,?p,InstanceURI)}`

The query for obtaining an instance data property and its value is:

SELECT `?p ?val` **WHERE** `{DataProperty(?p), PropertyValue(InstanceURI,?p,?val)}`

Although, the primordial idea of development a tool, such as the Query Ontology Interface, was to help in the migration process of the Archives Metadata into CIDOC-CRM, it can be used with other ontologies to explore their population individuals, as Figure 9 shows. In this figure, the instance of the Wine ontology (http://protege.stanford.edu/ontologies/win/win.owl), 'CortonMontarchetWhiteBurgon', is explored. It presents its object

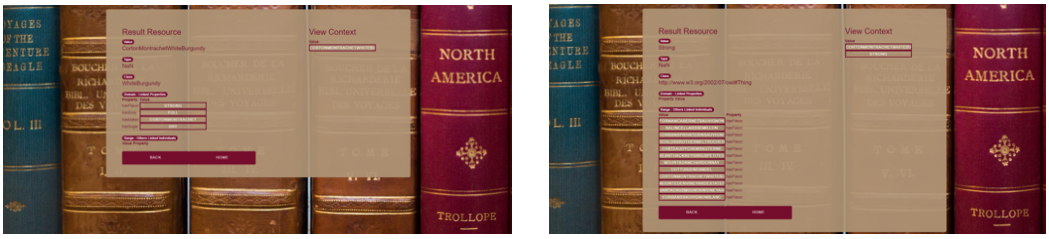

Fig. 9. Example of the Wine Ontology. (a) Instance 'CortonMontarchetWhiteBurgon'. (b) Instance 'Strong'.

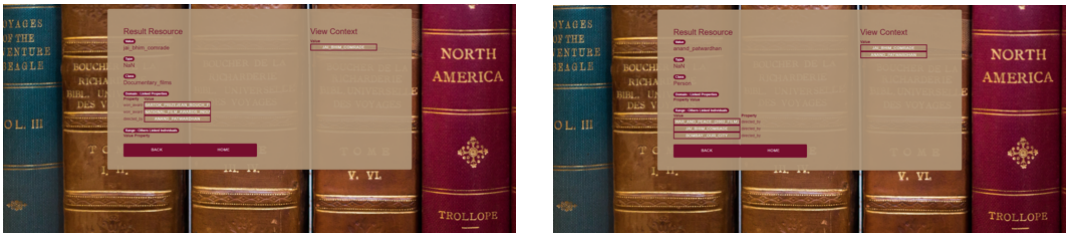

Fig. 10. Example of the WikiMovie Ontology. (a) Instance ''jai_bhim_comrade". (b) Instance 'anand_patwardhan'.

properties 'hasFlavor Strong'; 'hasBody Full'; 'hasMaker CortonMontarchet' and 'hasSugar Dry'. This instance does not have any inverse object properties instances in this ontology.

With the purpose to analyse and verify the interoperability of the Query Ontology Interface tool developed, another ontology population, the WikiMovie ontology (https://sites.google.com/site/ontoworks/ontologies), was tested and some exploration was made. This ontology describes information about Movies and their principal characteristics, such as directors, actors, awards, etc.. Figure 10 (a) shows that the instance 'jai_bhim_comrade' is categorized as a 'Documentary_films', won two awards, and was 'directed_by' 'anand_patwardhan'. About the exploration of the instance 'anand_patwardhan', it is possible to visualize, Figure 10 (b) that it is a 'Person' and the properties list all the movies 'directed_by' him. At the top right side of each layout, for instance the Figure 10 (b), it is possible to view the history path that led to the current layout exploration.

From these experiments, it is possible to verify that the exploration of an ontology instances is easily made through the Query Ontology Interface presented, regardless the ontology design, making the tool an useful one when exploring ontologies populations.

4. Conclusion and Future Work

In this paper, a Query Ontology Interface to explore a CIDOC-CRM population was presented. The interface was built with the purpose to help in the development of an automatic semantic migration prototype, based on Knowledge Discovery, from Digital Archive metadata to populate an ontology in CIDOC-CRM. The visualisation of the ontology instances allows the migration process developers to visualise the instances properties, and to debug the migration process and the migration representation model.

It was also presented the Migration Mapping Rules, along with its representation in a diagram view to explain the information that can be visualized in the interface. The interface has some features that were designed to explore CIDOC-CRM ontologies, such as the 'E_{55} Type' class, but it does not prevent the use with other ontologies as it was shown for two different ontologies. A tool such as Protegé enables users to explore an ontology population, but requires from users to be more experts since it is difficult to see an instance inverse object properties. To see an inverse property, the DL query has to be used for each inverse property. So the user must know the ontology and must be able to use the DL query. The proposed tool can be used by our partners archivists to explore the information even when they ignore ontology CIDOC-CRM details. The Interface tool was developed using state of the art tools, such as modeling and reasoners OWL-API with SPARQL-DL queries to obtain the exploration results.

In a close future, the interface shall be improved in some of the following aspects: add an option to hide the entities used to represent ternary relations, this is important in an ontology such as CIDOC-CRM that has a normalized representation for types over properties; add the possibility of loading a new ontology to explore its population; add the possibility of querying the ontology with DL queries.

Acknowledgements

This work is financed by National Funds through FCT - Foundation for Science and Technology I.P., within the scope of the EPISA project - DSAIPA / DS / 0023/2018 and NOVA LINCS (UIDB/04516/2020).

References

[1] Beretta, F., 2020. A challenge for historical research: making data fair using a collaborative ontology management environment (ontome). Semantic Web , 1–16.
[2] Beretta, F., Alamercery, V., 2019. Workflow for communal ontology management: aligning data models with OntoME, in: APOLLONIS Workshop "Historical content metadata", Centre for Cultural Informatics – Institute of Computer Science - FORTH, Heraklion, Greece.
[3] Bountouri, L., Gergatsoulis, M., 2011. The semantic mapping of archival metadata to the cidoc crm ontology. Journal of Archival Organization 9, 174–207.
[4] di Buono, M.P., Monteleone, M., Elia, A., 2014. How to populate ontologies, in: Métais, E., Roche, M., Teisseire, M. (Eds.), Natural Language Processing and Information Systems, Springer International Publishing, Cham. pp. 55–58.
[5] Cavalcanti, M.C., Pereira, F.D., Fusco, E., Mucheroni, M.L., 2017. Model of data extraction in the innovation environments of the state of são paulo based on semantic technologies, in: 14th Int Conf on Information Systems & Technology Management, CONTECSI USP, SP, Brazil.
[6] Fragkou, P., Kritikos, N., Galiotou, E., 2016. Querying greek governmental site using sparql, in: Proceedings of the 20th Pan-Hellenic Conference on Informatics, Association for Computing Machinery, New York, NY, USA.
[7] Horridge, M., Bechhofer, S., 2011. The owl api: A java api for owl ontologies. Semant. Web 2, 11–21.
[8] ICOM/CIDOC, 2020. Definition of the CIDOC Conceptual Reference Model. 7.0.1 ed., ICOM/CRM Special Interest Group.
[9] International Council on Archives, 2011. ISAD(G): general international standard archival description, Second Edition. Springer Nature BV.
[10] Kordjamshidi, P., Moens, M.F., 2015. Global machine learning for spatial ontology population. Journal of Web Semantics 30, 3–21.
[11] Kremen, P., Sirin, E., 2007. Sparql-dl implementation experience, in: Proceedings of the 4th OWLED Workshop on OWL: Experiences and Directions Washington.
[12] Leshcheva, I., Begler, A., 2020. A method of semi-automated ontology population from multiple semi-structured data sources. Journal of Information Science 0.
[13] Lohmann, S., Negru, S., Bold, D., 2014. The protégévowl plugin: Ontology visualization for everyone, in: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (Eds.), The Semantic Web: ESWC 2014 Sat Evt, Springer International Publishing, Cham. pp. 395–400.
[14] Lourdi, I., Papatheodorou, C., Doerr, M., 2009. Semantic integration of collection description: Combining cidoc-crm and dublin core collections application profile. D-lib Magazine - DLIB 15.
[15] Lubani, M., Noah, S.A.M., Mahmud, R., 2019. Ontology population: Approaches and design aspects. Journal of Information Science 45, 502–515.
[16] Makki, J., 2017. Ontoprima: A prototype for automating ontology population. International Journal of Web/Semantic Technology (IJWesT) 8.
[17] Marlet, O., Francart, T., Markhoff, B., Rodier, X., 2019. OpenArchaeo for Usable Semantic Interoperability, in: ODOCH 2019 @CAiSE 2019.
[18] Maynard, D., Li, Y., Peters, W., 2008. Nlp techniques for term extraction and ontology population.
[19] Meghini, C., Bartalesi, V., Metilli, D., 2021. Representing narratives in digital libraries: The narrative ontology, in: Bikakis, A., Hyvonen, E., Jean, S., Markhoff, B., Mosca, A. (Eds.), Semantic Web. IOS Press. volume 12, p. 241 – 264.
[20] Meghini, C., Doerr, M., 2018. A first-order logic expression of the cidoc conceptual reference model. International Journal of Metadata, Semantics and Ontologies 13, 131–149.
[21] Melo, D., Rodrigues, I.P., Koch, I., 2020a. Knowledge discovery from isad, digital archive data, into archonto, a cidoc-crm based linked model, in: Proceedings of the 12th International Joint Conference on Knowledge Discovery, KEOD - Volume 2, INSTICC. SciTePress. pp. 197–204.
[22] Melo, D., Rodrigues, I.P., Varagnolo, D., 2020b. Installation unit - registos de baptismos. doi:10.17632/wx7v7rmg7h.2.
[23] Myers, D., Quintero, M.S., Dalgity, A., Avramides, I., 2016. The arches heritage inventory and management system: a platform for the heritage field. Journal of Cultural Heritage Management and Sustainable Development .
[24] Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., Zavitsanos, E., 2011. Ontology Population and Enrichment: State of the Art. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 134–166.
[25] Ramalho, J.C., Ferreira, J.C., 2004. Digitarq: creating and managing a digital archive, in: Building Digital Bridges: Linking Cultures, Commerce and Science: 8th ICCC/IFIP International Conference on Electronic Publishing held in Brasília - ELPUB 2004, Brazil, June, 2004.
[26] Sirin, E., Parsia, B., 2007. Sparql-dl: Sparql query for owl-dl, in: 3rd Workshop on OWL: Experiences and Directions.
[27] Theodoridou, M., Doerr, M., 2001. Mapping of the encoded archival description dtd element set to the cidoc crm. Technical Report FORTHICS/TR-289. FORTH .
[28] Vitali, S., 2004. Authority control of creators and the second edition of isaar (cpf), international standard archival authority record for corporate bodies, persons, and families. Cataloging & classification quarterly 38, 185–199.