# Building the Museum of the Person from RDF Triples and SPARQL

**Cristiana Araújo**[1]**, Ricardo G. Martini**[1]**,**
**Pedro Rangel Henriques**[1]**, José João Almeida**[1]

[1]Algoritmi Research Centre, Department of Informatics
University of Minho - Gualtar - 4710-057, Braga, Portugal

decristianaaraujo@hotmail.com, rgm@algoritmi.uminho.pt,

{prh,jj}@di.uminho.pt

***Abstract.*** *The Museum of the Person (MP) is a virtual museum that aims to show life stories of people, whether they are famous or anonymous. Its repository contains a collection of interviews, in XML (eXtensible Markup Language). Each interview tells the story of a person's life, describing events and other special situations that person has participated. The main objective is to build web pages that carry out the museum exhibition rooms. To feed the exhibition rooms it is necessary to (automatically) extract the information included in the repository. Initially was constructed an ontology specific to the repository of the Museum of the Person, designated OntoMP. Then we adopted a standard to create ontologies for museums, CIDOC-CRM (CIDOC - Conceptual Reference Model) refined with FOAF (Friend of a Friend) and DBpedia to represent OntoMP. This article aims to discuss the construction of virtual rooms using a TripleStore to store the repository information and SPARQL technology (SPARQL Protocol and RDF Query Language) to extract information from the XML repository.*

***Resumo.*** *O Museu da Pessoa (MP) é um museu virtual que visa exibir histórias de vida de pessoas, sejam elas famosas ou anónimas. O seu repositório contém uma coleção de entrevistas, no formato XML (eXtensible Markup Language). Cada entrevista relata a história de vida de uma pessoa, referindo eventos e outras situações particulares em que a pessoa tenha participado. O objectivo principal consiste em construir páginas web que realizem as salas de exposição do museu, mas para alimentar as salas de exposição é necessário extrair automaticamente as informações incluídas no repositório. Inicialmente construiu-se uma ontologia especifica para o repositório do Museu da Pessoa, designada OntoMP. Definida a ontologia adotou-se um padrão de ontologias para museus, CIDOC-CRM (CIDOC - Conceptual Reference Model) refinado com FOAF (Friend of a Friend) e DBpédia, para representar OntoMP. Este artigo tem como objectivo mostrar a contrução das salas virtuais, utilizando um TripleStore para armazenar o repositório de informação e a tecnologia SPARQL (SPARQL Protocol and RDF Query Language) para extrair as informações do repositório XML.*

## 1. Introduction

The Museum of the Person (MP) is a global network that relates individuals and groups through authoring and sharing their life stories. The Museum branches are located in Brazil, Portugal, USA, and Canada[1]. The MP aims at recording, preserving, and transforming in information the life stories of any person of the society. Its repository contains a set of interviews containing life stories of persons (anonymous or not) in XML (eXtensible Markup Language) format. This repository can be considered a Cultural Heritage asset.

Archives, libraries, and museums are figuring out new ways to export their data in triples to disseminate broadly their assets (Cultural Heritage) [Marden et al. 2013]. So triples can be explored through websites containing features for viewing, interacting, and navigating over data.

Looking in this direction, we propose in this paper, Sections 3 and 4, an architectural approach[2] to deal with this issue and turn data of the Museum of the Person available to any visitor.

We present our contribution in terms of preparing data to be available on the web and to build web pages as exhibition rooms to disseminate the prepared data. To prepare the MP assets to accomplish the objectives of this work, we have developed an ontology called OntoMP, that is detailed in Section 2. After data be prepared, we have used SPARQL to disseminate the life stories via exhibition rooms at a virtual museum, which is presented in Section 5.

## 2. OntoMP – The Ontology for Museum of the Person

After an intensive analysis of the Museum of the Person's assets (original and edited interviews, BI's, photos, DTD's and *Thesaurus*), it was possible to identify concepts and relations involved in the stories of people's lives [Simões and Almeida 2003, Almeida et al. 2001]. With these concepts and relationships we could draw an ontology, which was called *OntoMP: Original Design*. Once built we found that some elements could be added to the ontology, given the information contained in the MP's assets. The elements added were: marital status, gender, literacy, political party, first communion, death, baptism and photos. The next phase was to validate the proposed ontology. For that, we populated the ontology with several real life stories taken from the assets of the Museum of the Person.

Once OntoMP was validated, we decided to describe it using a standard ontology defined for museums, CIDOC-CRM (CIDOC Conceptual Reference Model). CIDOC-CRM is a formal ontology that was created to assist in the integration, mediation and exchange of information in the area of Cultural Heritage. CIDOC-CRM is an Event-based ontology, and therefore it should contain *Time-Spans* and *Places* related with each event. The core of CIDOC-CRM is based on seven concepts: *Temporal Entities*, *Events*, *Actors*, *Time-Spans*, *Conceptual Objects*, *Physical Things*, and *Places*. Notice that, *Actors* and *Conceptual Objects* or *Physical Things* should also be related with *Event* [ICOM/CIDOC 2015].

---

[1]Available at: `http://www.museumoftheperson.org/about/`
[2]Note that this approach is generic, i.e., it serves for any kind of museums.

The transformation of *OntoMP: Original Design* in CIDOC-CRM was a straight-forward process; the original concepts were expressed as events and associated concepts, and the original relations were mapped into the correspondent in CIDOC-CRM. However, we found that some properties related with person could not be expressed in CIDOC-CRM in a simple and natural form. So we decided to explore the combination with FOAF (Friend of a Friend) and DBpedia, since both contain a vocabulary specific to describe individuals, their activities and their relations with other people and objects [Allemang and Hendler 2011].

FOAF ontology describes two areas of digital identity information: biographical and social network information [Al-Mukhtar and Al-Assafy 2014].

DBpedia ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. DBpedia knowledge base covers various fields, such as geographic information, people, businesses, online communities, movies, music, books and scientific publications, among others [Dbpedia 2016].

After this investigation, we refined CIDOC-CRM adding FOAF and DBpedia concepts and properties. Regarding FOAF, we imported *gender* property, person names (*name*, *givenName*, *familyName* and *nick*) and person-image relations (*depicts* and *depiction*). From DBpedia we picked up properties like *religion*, *profession*, *education*, *party* and *spouse*.

After the refinement of CIDOC-CRM ontology with FOAF and DBpedia elements, we got a simpler notation (descriptions became less verbose); moreover the original was enriched conceptually, this is more details about people's stories can be included in the knowledge base. The final OntoMP was once again instantiated with concrete data extracted from the real life stories. It was possible to validate it once more.

In Figure 1 we show an instance of the ontology created with data extracted from Maria Cacheira interview. Below we describe the CIDOC-CRM, FOAF and DBpedia fragment reproduced.

A person *(E21 Person)*, *gender* Female, *name* Maria Alice Rodrigues Cacheira (decomposed in *givenName* Maria Alice and *familyName* Rodrigues Cacheira), *participated in (E5 Event)* that is her birth *(E67 Birth)*. This event occurred at a *(E52 Time Span)*—that *is identified by (P78)* 1946-10-08, an *(E50 Date)*—and at a *(E53 Place)*—that *is identified by (P87)* Afurada an *(E44 Place Appellation)*.

This person *(E21 Person)* is *depicted* in the photo *(E38 Image)*. This photo *is identified by (P1)* 090-F-01.jpg *(E41 Apellation)*, *has note (P3)* Maria Alice Rodrigues Cacheira, *refers to (P67)* Maria Alice Rodrigues Cacheira *(E55 Type Description)*, and was taken in a *(E52 Time Span)*—that *is identified by (P78)* 2001-12-07, an *(E50 Date)*—and at a *(E53 Place)*—that *is identified by (P87)* Junta de Freguesia da Afurada, an *(E44 Place Appellation)*.

A person *(E21 Person)* has *education* "Sabe ler e escrever (4ª classe)", professes the *religion* "Católica" and has *profession* "Peixeira e Empregada de limpeza".

In this fragment of Maria Cacheira's life story other concepts can be identified. All these concepts, that characterize a *(E21 Person)*, are represented in CIDOC-CRM

version, as *(E55 Type)*. For example, *(E21 Person) has type (P2)* "Viúva" *(E55 Type Marital Status)*.
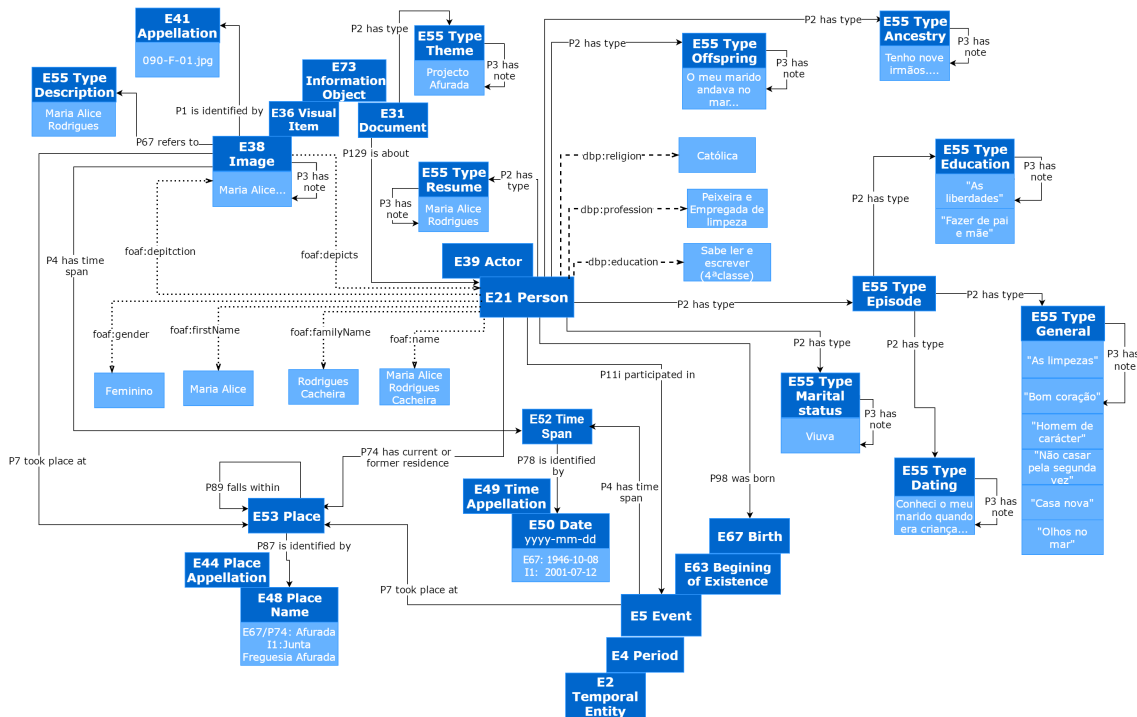


**Figure 1. An instance of CIDOC-CRM / FOAF / DBpedia version of OntoMP for Maria Cacheira life story (fragment)**

The person's properties imported from FOAF (above identified) are emphasized in Figure 1 using dotted line. Similarly, DBpedia properties used are enhanced as dashed line.

This CIDOC-CRM ontology enriched with FOAF and DBpedia elements can describe appropriately the knowledge repository of the Museum of the Person. The next step is to present the architecture of a system that builds the virtual museum from the document collection.

## 3. Building MP on the top of OntoMP

The first step to implement the Museum of the Person was the design of a general architecture that establishes the abstract blocks that are mandatory to solve our problem: *extract the information from the XML document collection and build the exhibition rooms of the virtual museum on the web*. According to the detailed description presented in [Araújo et al. 2016], this general architecture comprises: the repository; the Ingestion Function [M1] responsible for getting and processing the input data; a Data Storage [DS] that is the digital data archive; an Ontology to map and link the concepts with the objects stored in [DS]; the Generator [M2] to extract data from [DS] and manage the information that will be displayed in Virtual Learning Spaces [VLS].

Figure 2 depicts this architecture that has clearly to main parts, on the left side the archive one, and on the right side the exhibition one (VLS).
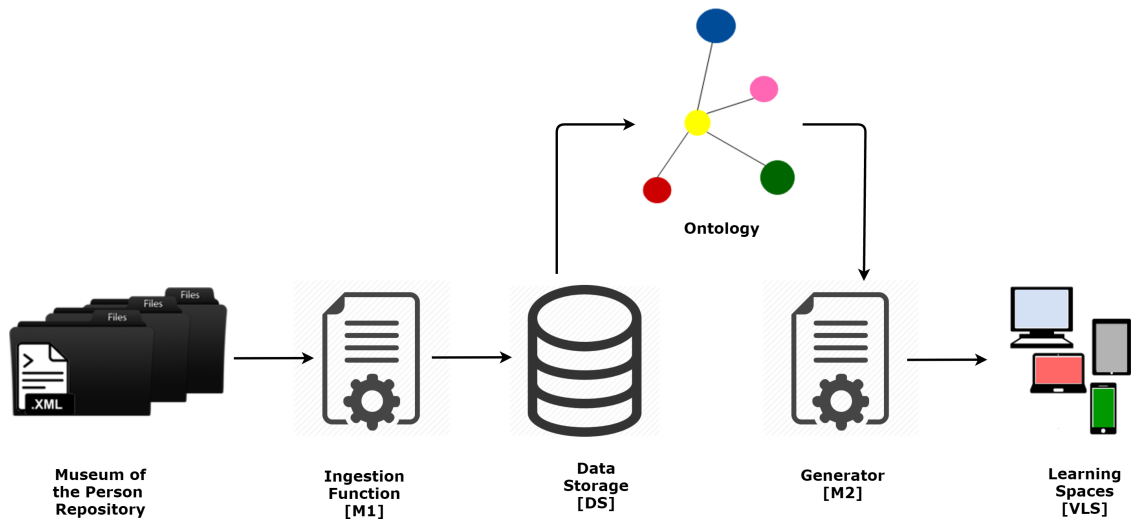
**Figure 2. General Architecture to build the Museum of the Person**

This general architecture has two possible refinements, which are dependent on the Data Storage [DS], as discussed in [Araújo et al. 2016]. In approach 1 the [DS] is a *TripleStore*, while in approach 2 the [DS] is a *Relational Database*. The choice of the storage technique completely determines the implementation of part one [M1] and the connection between the ontology and the store (for more details, please see the above referred paper [Araújo et al. 2016]).

As we chose the first approach to go on with the development, in this paper we only show in detail the stages of approach 1. The detailed discussion of both parts of the selected approach will be distributed by the two following sections.

## 4. Storing OntoMP as RDF Triples

In this approach the Ingestion Module [M1], part 1 of our general architecture, will transform the input XML documents into RDF (Resource Description Framework) triples [McBride 2004]. Figure 3 details the first module that is composed of three blocks:

- *Parser and Semantic Checker* that reads the repository documents (annotated in XML) and extracts the relevant data, checking the syntactic and semantic consistency;
- *Ontology Extractor* that identifies in the extracted data the concepts and relations that belong to the ontology creating in this way an instance of the abstract ontology (in another words, this component populates the ontology);
- *Triple Generator* that converts automatically the ontology triples into RDF (Resource Description Framework) triples in notation appropriated to be stored in the [DS] chosen.

In this approach there is no need to create an explicit mapping between the domain ontology and the data extracted from the repository, because the data storage archives directly the ontology triples.

At this stage, analysis and extraction of information from documents making up the ontology instance is still done manually. This means that the Parser and Semantic
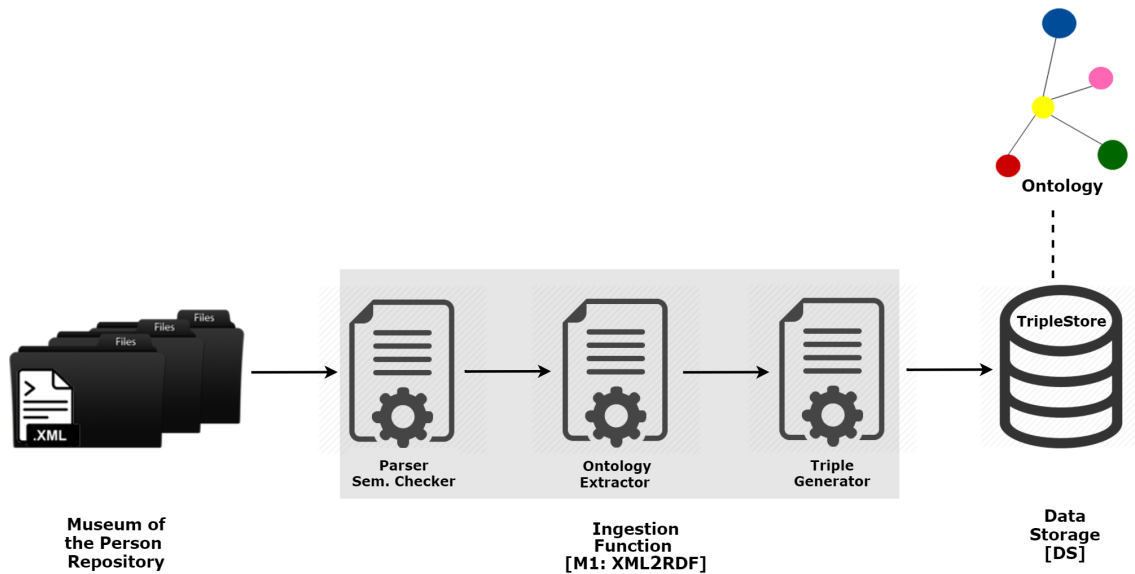
**Figure 3. Module [M1] in Approach 1**

Checker and the Ontology Extractor tools have not yet been implemented (at this moment we are working).

As previously said, to implement MP we decided to use a TripleStore to archive the ontology triples, as explained above. Although there are other notations for ontology description, we chose RDF because we use CIDOC-CRM, FOAF and DBpedia that are described, in its original form, in RDF. Listing 1 shows an excerpt of the RDF triples built.

**Listing 1. Fragment of the RDF Triples for Maria Cacheira life story**

```
1  <!-- Description Interviewed 1 -->
2  <rdf:Description rdf:about="&ecrm;Interviewed_1">
3        <rdf:type rdf:resource="&ecrm;E21_Person"/>
4        <rdf:type rdf:resource=
      "http://dbpedia.org/ontology/Person"/>
5        <rdf:type rdf:resource="&foaf;Person"/>
6
7        <foaf:firstName rdf:datatype="&xsd;string">Maria
      Alice</foaf:firstName>
8        <foaf:name rdf:datatype="&xsd;string">Maria Alice
      Rodrigues Cacheira</foaf:name>
9        <foaf:familyName rdf:datatype="&xsd;string">
      Rodrigues Cacheira</foaf:familyName>
10       <P98i_was_born rdf:resource="&ecrm;B1"/>
11
12       <foaf:gender rdf:datatype="&xsd;string">
      Feminino</foaf:gender>
13       <foaf:depiction rdf:resource=
      "&ecrm;I1_Interviewed_1"/>
14
15       <dbp:profession rdf:datatype="&xsd;string"> Peixeira
```

```
16      e empregada de limpeza</dbp:profession>
            <dbp:religion rdf:datatype="&xsd;string">
        Catolica</dbp:religion>
17          <dbp:education rdf:datatype="&xsd;string"> Sabe ler
        e escrever (quarta classe)</dbp:education>
        </rdf:Description>
18
19  <!-- Event Birth Interviewed 1  (B1) -->
20  <rdf:Description rdf:about="&ecrm;B1">
21          <rdf:type rdf:resource="&ecrm;E67_Birth"/>
22          <P98_brought_into_life rdf:resource=
        "&ecrm;Interviewed_1"/>
23          <P4_has_time-span rdf:resource="&ecrm;TS1"/>
24          <P7_took_place_at rdf:resource="&ecrm;PL1"/>
25  </rdf:Description>
26
27  <!-- Description Photo Interviewed 1 (I1) -->
28  <rdf:Description rdf:about="&ecrm;I1_Interviewed_1">
29          <rdf:type rdf:resource="&ecrm;E38_Image"/>
30          <rdf:type rdf:resource="&foaf;Image"/>
31          <foaf:depicts rdf:resource="&ecrm;Interviewed_1"/>
32          <P67_refers_to rdf:resource=
        "&ecrm;I1_Description_Interviewed_1"/>
33          <P3_has_note rdf:datatype="&xsd;string"> Maria Alice
        Rodrigues Cacheira</P3_has_note>
34          <P1_is_identified_by rdf:resource=
        "&ecrm;090-F-01.jpg"/>
35          <P4_has_time-span rdf:resource="&ecrm;TS7"/>
36          <P7_took_place_at rdf:resource="&ecrm;PL8"/>
37  </rdf:Description>
```

The triple fragment shown in Listing 1 contains information about life story of Maria Cacheira. The first section (line 1-17) displays the biographic information about Maria Cacheira, as name (first, last name and full name), birth, sex, photo, profession, religion, and education. The second section (line 19-25) describes the birth event of Maria Cacheira, date and place of it. The last section (line 27-37) contains specific information about the photo of the interviewed, such as description, legend, file, date and place.

To clarify the organizational form of the triples created, Figure 4 shows Turtle notation . This ontology representation is simpler and so it becomes easier to read the ontology instance and understand the number of triples and how they are formed.

In Figure 4 we can see that there are 12 triples each one containing three parts: subject, predicate and object. A triple example is: *Interviewed_1* (subject) *foaf: name* (predicate) *"Maria Alice Rodrigues Cacheira" ^ ^ xsd: string* (object). Notice that the remaining triples in this fragment follow the same logic and in all of them the subject is *Interviewed_1*.

After building the RDF triples manually, it was necessary to validate them to ensure that the very long textual description produced contained no errors. For this, we used

```
:Interviewed_1  a              rdf:Description , dbp:Person ,
                               :E21_Person , foaf:Person ;

       foaf:name               "Maria Alice Rodrigues Cacheira"^^xsd:string ;
       foaf:firstName          "Maria Alice"^^xsd:string ;
       foaf:familyName         "Rodrigues Cacheira"^^xsd:string ;
       :P98i_was_born          :B1 ;

       foaf:gender             "Feminino"^^xsd:string ;
       foaf:depiction          :I1_Interviewed_1 ;

       dbp:profession          "Peixeira e empregada de limpeza"^^xsd:string ;
       dbp:religion            "Católica"^^xsd:string ;
       dbp:education           "Sabe ler e escrever (4ª Classe)"^^xsd:string .
```

**Figure 4. Maria Cacheira life story triples in Turtle notation (fragment)**

the online tool RDF Validator W3C [3], which checks the consistency of the RDF triples and displays them in a table with three columns 'subject, predicate and object'. After loading our RDF file we got as feedback the information *"VALIDATION RESULTS: Your RDF document validated successfully"* that is just we want to get from that tool. The huge triples table was also useful for a quick visual reference.

After validated successfully by RDF Validator, the next step was to store them in a data set. For this we used Apache Jena TDB, which is an RDF database.

TDB is a component of Jena (free and open source Java framework for building Semantic Web and Linked Data applications) for RDF storage and query, and can be used as a high performance RDF store on a single machine. A TDB store can be accessed and managed with the provided command line scripts and via the Jena API. When accessed using transactions a TDB dataset is protected against corruption, unexpected process terminations and system crashes. Apache Jena Fuseki component provides a SPARQL server to be used with TDB [JENA 2016b].

In the next section, we will describe the second part of our architectural approach 1, and how the RDF triples stored in Apache Jena TDB is queried to produce the virtual exhibition rooms .

## 5. Querying OntoMP

As previously stated, the modules that compose the second part of our system will retrieve information from the RDF TripleStore to create the museum web pages. This task is performed by the Virtual Learning Spaces Generator [M2].

The [VLS] Generator needs to send queries and process the returned data, to display in the Virtual Learning Spaces [VLS] the information stored in the TripleStore [DS]. Figure 5 shows the second module [M2] (the Generator) that is composed of two blocks:

- *SPARQL Endpoint* that receives and interprets the SPARQL queries, accesses the TripleStore and returns the answers;
- *Query Processor* generates the SPARQL queries according to the exhibition room requirements, sends them to the SPARQL Endpoint and, after receiving the answer, combines the returned data to set up the Virtual Learning Spaces [VLS].
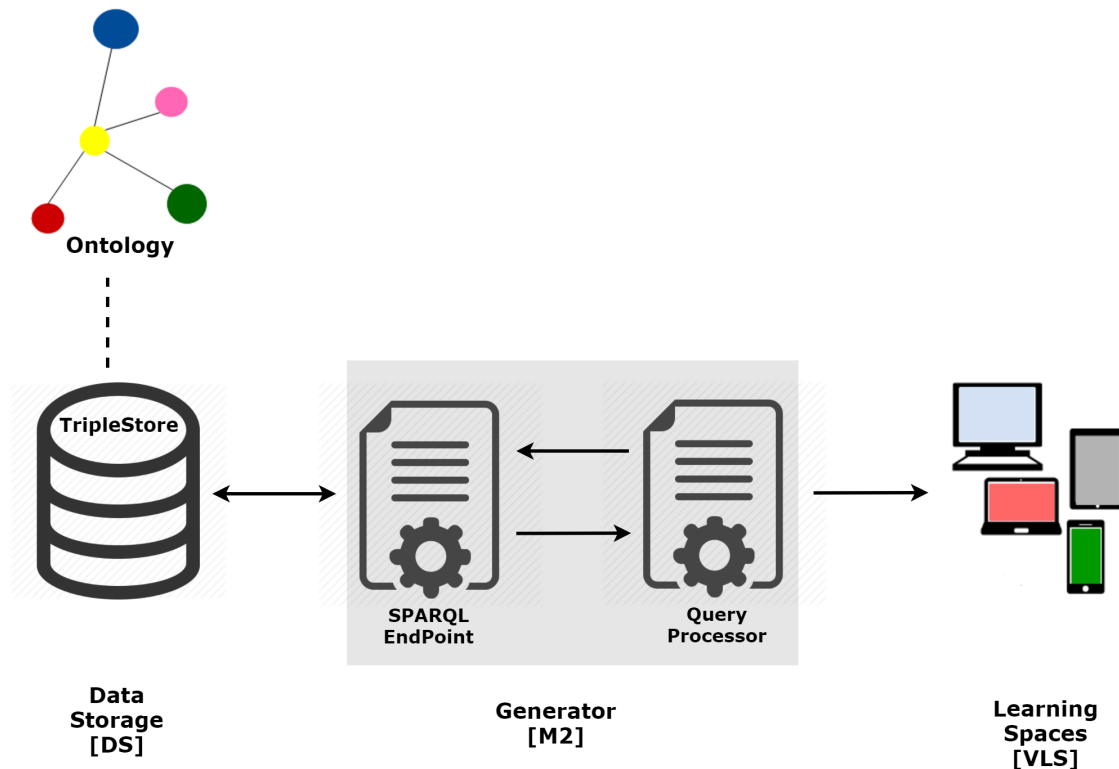
**Figure 5. Module [M2] in Approach 1**

As already mentioned, to run SPARQL queries sent by the Query Processor it is necessary to resorted to a SPARQL Endpoint. The SPARQL endpoint used was Apache Jena Fuseki (version 2.0).

Apache Jena Fuseki is a SPARQL server, that can run as a operating system service, as a Java web application (WAR file), and as a standalone server. It provides security (using Apache Shiro) and has a user interface for server monitoring and administration. Fuseki is tightly integrated with TDB to provide a robust, transactional persistent storage layer, and incorporates Jena text query and Jena spatial query. It can be used to provide the protocol engine for other RDF query and storage systems [JENA 2016a].

Some queries were built to test if we could get the answers required. Examples of the queries issued are: 1. Interviewed and respective Project; 2. Interviewed and their photos; 3. Interviewed, his profession, marital status and spouse's name; 4. Interviewed and his events; 5. Interviewed by sex and residence; 6. Interviewed by type of episode; 7. Total Interviewed by Project; 8. Life story of Interviewed. All these queries were run successfully in Fuseki web interface.

Listing 2 displays the SPARQL query: *Interviewed and his photos*.

**Listing 2. Query SPARQL: Interviewed and his photos**

```
1  PREFIX : <http://erlangen-crm.org/150929/>
2  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
3  PREFIX dbp: <http://dbpedia.org/ontology/>
4  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

---

³https://www.w3.org/RDF/Validator/

```
 5  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
 6
 7  SELECT DISTINCT ?nome  ?ficheiro ?personDesc ?legenda
        ?dataPhoto ?placePHOTO
 8
 9  WHERE {
10    ?pessoa a :E21_Person;
11      :P129_is_subject_of ?doc;
12      foaf:name ?nome;
13      foaf:depiction ?fotos.
14        ?fotos :P1_is_identified_by ?ficheiro.
15      OPTIONAL{
16          ?fotos :P3_has_note ?legenda.}
17      OPTIONAL{
18          ?fotos :P67_refers_to ?personDesc.
19              ?personDesc :P3_has_note ?personDescricao.}
20      OPTIONAL{
21          ?fotos :P4_has_time-span ?data.
22              ?data :P78_is_identified_by ?dataPhoto.}
23      OPTIONAL{
24          ?fotos :P7_took_place_at ?placePhoto.
25              ?placePhoto :P87_is_identified_by ?placePhoto1.
26                  ?placePhoto1 :P3_has_note ?placePHOTO.}
27
28  }ORDER BY ?nome ?ficheiro
```

The code block between lines 10 and 14 of Listing 2 is intended to search for all respondents *(E21_Person)* and their photos in which they are represented *(foaf:depiction)*. The property *foaf:name* describes the full name of each interviewed.

We found that not all photos contain all the items we wish to retrieve. Therefore, in the query we use the argument *OPTIONAL* to cope properly with that variability in the legend (lines 15-16), in the description (lines 17-19), in the date (lines 20-22) and in the photo place (lines 23-26).

For more information on the results and executed queries, please see: `http://npmp.epl.di.uminho.pt`.

We created a Python script that generates SPARQL queries according to the requirements of each exhibition room, sends them to the Fuseki (SPARQL Endpoint) and after receiving the answer, combines the data returned to configure the Virtual Learning Spaces [VLS]. This Python script also includes HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) to create and format the web page. We are aware of the computational time needed to process and answer SPARQL queries. However, at moment, the size (the number of ontology triples) of our datastore is not big enough to pose any real execution problem. We will measure it carefully when the ontology instances increase significantly and after facing real performance constraints we will think on a solution, maybe using a cache to memorize queries and respective answers.

Finally, the web pages were built (Virtual Learning Spaces) to exhibit the life

stories that are the objects of the Museum of the Person. Figure 6 displays the Museum of the Person entry page. Figure 7 displays the page where the museum visitor can perform the SPARQL queries. Figure 8 displays the response to the query: 2. Interviewed and his photos.
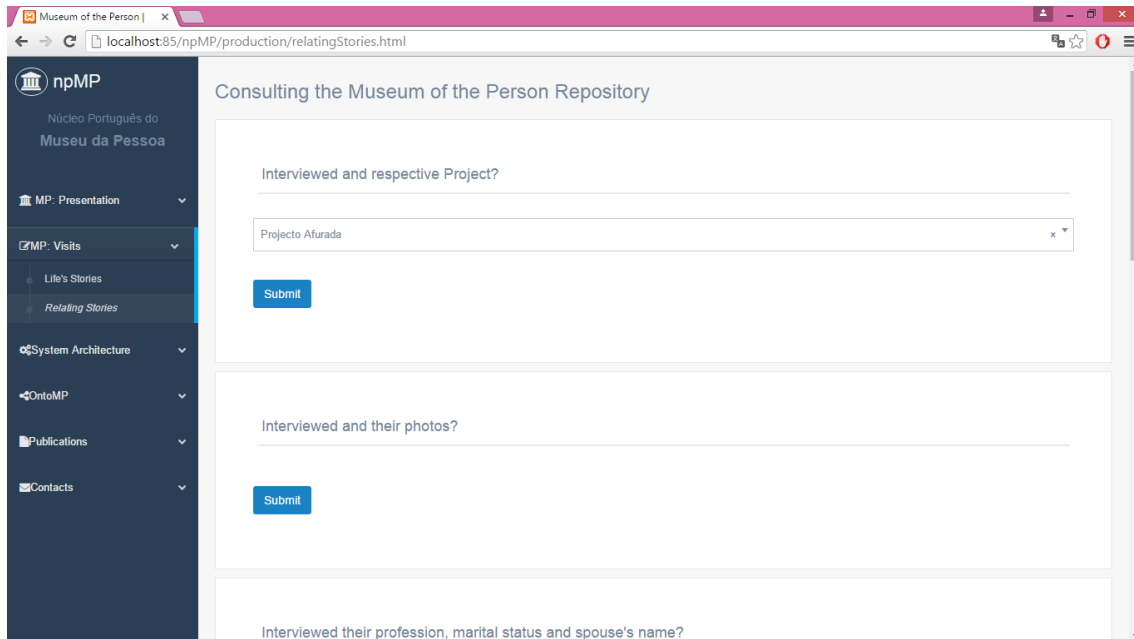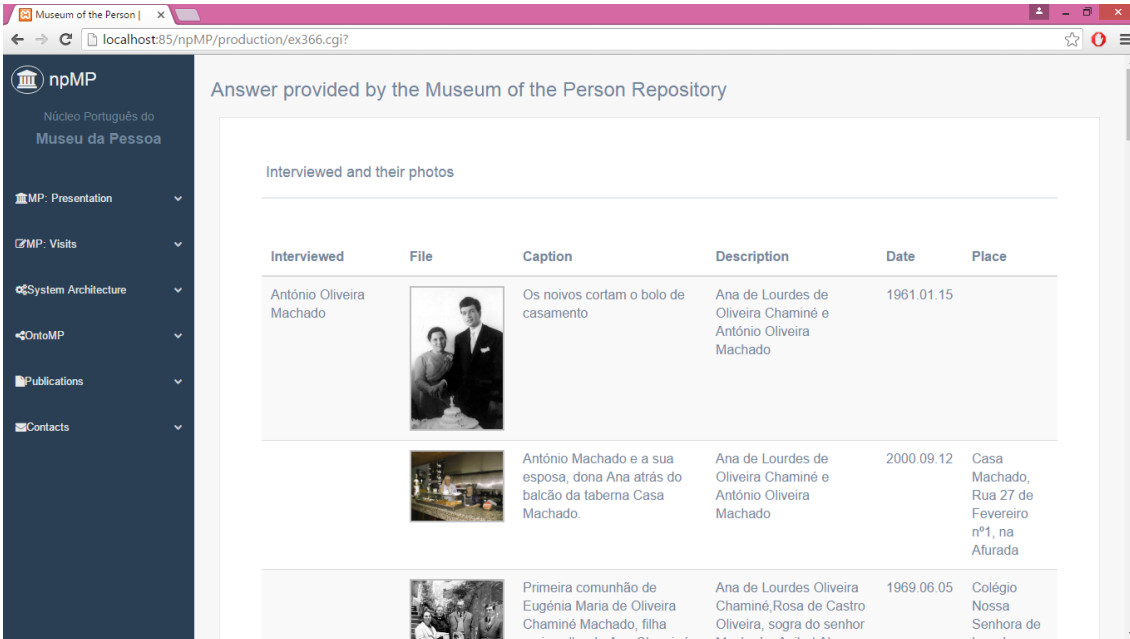


**Figure 6. Home Page Museum of the Person**



**Figure 7. Consulting the Museum of the Person Repository**

**Figure 8. Response to the SPARQL query:** *Interviewed and their photos*

In this approach, each Virtual Learning Space (a museum's exhibition room) is built fulfilling a web page template with the concrete data retrieved from the data store. Notice that menus and tiltes of the npMP website are in English (although the pages content is, of course, in Portuguese). This was a design decision (easily changeable) to allow foreigner people to access the museum pages and to learn something from them— actually the website contains much more information than just the lifestories (displayed in the museum exhibition rooms or virtual learning spaces).

## 6. Conclusion

This project aims at buiding in the Portuguese branch of the Museum of the Person to display the stories of people's lives, whether they are famous or anonymous. This allows perpetuate their life stories and rebuild the social universe of a city or a country in the epoch they lived.

We presented in this paper a CIDOC-CRM ontology combined with FOAF and DBpedia that was created from the concepts and relations identified after a careful analysis of the documents of the Museum of the Person. This ontology stores in RDF triples all life stories that make up the asset.

To extract the desired information about the life histories archived in the triple-store, SPARQL queries were created. This information returned as the answer to the queries is shown in the web pages, called Virtual Learning Spaces, created specifically for this project.

As future work we will implement the *Parser and Semantic Checker* to automatically read the repository documents and extract relevant data about people's lives, checking the document's syntactic and semantic consistency. We also want to automate the ontology population, in other words, create the *Ontology Extractor* to identify in the extracted data the concepts and relations that belong to the ontology creating in this way an instance of the abstract ontology.

## 7. Acknowledgment

## References

Al-Mukhtar, M. M. A. and Al-Assafy, A. T. A. (2014). The implementation of foaf ontology for an academic social network. *International Journal of Science, Engineering and Computer Technology*, 4(1):10.

Allemang, D. and Hendler, J. (2011). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier Science.

Almeida, J. J., Rocha, J. G., Henriques, P. R., Moreira, S., and Simões, A. (2001). Museu da Pessoa – arquitectura. In *Encontro Nacional da Associação de Bibliotecários, Arquivista e Documentalistas, ABAD'01*. BAD.

Araújo, C., Martini, R. G., Henriques, P. R., and Almeida, J. J. (2016). Architectural Approaches to build The Museum of the Person. In Rocha, Á., Reis, L. P., Cota, M. P., Suárez, O. S., and Gonçalves, R., editors, *Sistemas y Tecnologías de Información — Atas da 11ª Conferência Ibérica de Sistemas e Tecnologias de Informação*, volume Vol. I — Artículos de la Conferencia, pages 383–388. AISTI–Associação Ibérica de Sistemas e Tecnologias de Informação.

Dbpedia (2016). Ontology. http://wiki.dbpedia.org/. Accessed: 2016-06-15.

ICOM/CIDOC (2015). Definition of the CIDOC Conceptual Reference Model. Technical report, ICOM/CIDOC.

JENA, A. (2016a).    Apache Jena Fuseki.    `https://jena.apache.org/documentation/fuseki2/index.html`. Accessed: 2016-06-01.

JENA, A. (2016b). TDB. `https://jena.apache.org/documentation/tdb/index.html`. Accessed: 2016-06-01.

Marden, J., Li-Madeo, C., Whysel, N., and Edelstein, J. (2013). Linked Open Data for Cultural Heritage: Evolution of an Information Technology. In *Proceedings of the 31st ACM International Conference on Design of Communication*, SIGDOC '13, pages 107–112, New York, NY, USA. ACM.

McBride, B. (2004). *The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS*, pages 51–65. Springer Berlin Heidelberg, Berlin, Heidelberg.

Simões, A. and Almeida, J. J. (2003). Histórias de Vida + Processamento Estrutural = Museu da Pessoa. In *XATA 2003 — XML: Aplicações e Tecnologias Associadas*, page 16, Braga, Portugal. UM.