Automatizing Ontology Population to drive the navigation on Virtual Learning Spaces

Cristiana Araújo
Algoritmi Research Centre, Department of Informatics
University of Minho - Gualtar - 4710-057, Braga, Portugal
decristianaaraujo@hotmail.com

Pedro Rangel Henriques
Algoritmi Research Centre, Department of Informatics
University of Minho - Gualtar - 4710-057, Braga, Portugal
prh@di.uminho.pt

Ricardo G. Martini
Algoritmi Research Centre, Department of Informatics
University of Minho - Gualtar - 4710-057, Braga, Portugal
rgm@algoritmi.uminho.pt

Abstract — **The creation of Learning Spaces on the Web, like the exhibition rooms of virtual museums, supported by an ontology that enables a conceptual navigation over the learning objects exposed, is an hard and complex task but of uttermost importance for the success of the knowledge acquisition process. In our opinion, the creation must be systematic and reusable from case to case, based on the query of the ontology instances that describe the museum assets. We will discuss how the ontology definition drives the way SPARQL (SPARQL Protocol and RDF Query Language) queries extract information from the TripleStore to be prepared for visualization. However, to enable this approach, we need to populate the ontology in an automatic way, extracting the data from the annotated documents in the institution repository. We intend to show how that process can be implemented using the Museum of the Person (MP) as a case-study, describing the XML2RDF tool developed. To illustrate the complete approach proposed we will include a guided visit to the exhibition rooms of MP created according to that proposal and by our tools.**

Keywords – Virtual Learning Spaces; Ontology; XML; RDF; SPARQL.

I. Introduction

In the context of our work, a virtual museum is one that has no physical existence (is not located in a building and has no physical objects to show), and displays in its exhibitions rooms objects collected from a digital repository that constitutes the museum assets. Exhibition rooms are Web pages and the visitor access the collection navigating over a browser [1].

To create such virtual rooms on the Web (usually we call them *learning spaces*), it is necessary to query the repository's digital storage, and to process (transform and relate) the returned information before publishing it. The discussion of the approach proposed to implement this process is one of our goals in the present article.

Sometimes the storage is a relational database, other times it is a collection of annotated documents.

In our research group, we have experience in coping with both cases, aiming at the implementation of generic and efficient tools able to extract the necessary data and relations automatically. We also investigated how to build the virtual museum web pages in a systematic way that can be easily adapted from one project to another.

In this article, we consider the second case, annotated documents, and construct a text filter capable of automatically create triples that will populate the museum's ontology. This text filter translates XML (eXtensible Markup Language) documents into RDF (Resource Description Framework) notation. As a case study, to illustrate the implementation of this process and its successful application, we will use the assets of the *Museum of the Person* (MP) [2] [3] [4].

Figure 1 depicts our point of view concerning the global process: from the digital repository to the Virtual Learning Spaces, via a domain ontology [5].

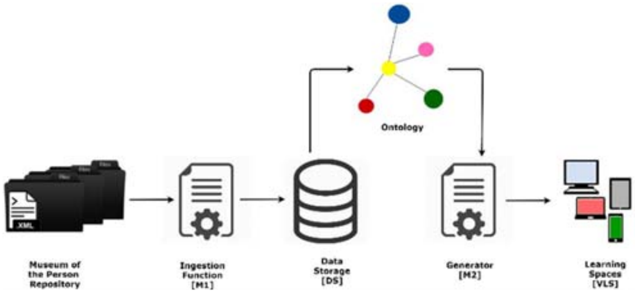


Figure 1. General Approach to build Virtual Learning Spaces

This architecture comprises: the repository; the Ingestion Function [M1] responsible for reading the annotated documents, extracting and preparing the data, and store the information gathered; a Data Storage (DS) that contains the ontology instances; an Ontology that describes the knowledge domain linking the concepts through a set of relations; the Generator [M2] to receive and interpret the requests for information, access the DS and return the answers that are combined to set up Virtual Learning Spaces (VLS) [6] [7].

In Section 2, we discuss the design and the development of the text filter, named *XML2RDF translator*, whose function is to transform XML documents into RDF triples; this is one of the contributions of the present paper. The creation of Virtual Learning Spaces (VLS) and how we extract the information stored in the ontology to display on the Web (on the VLS), a second contribution, is presented in Section 3. In Section 4 a guided visit to Museum of the Person is shown, as a case study to test the translator built. Finally, Section 5 presents the conclusion and directions for future work.

II. XML2RDF Translator

To design the Ingestion Function [M1] it is important to know that the input is a structured collection of tagged data and the output will be sequence of the <subject, predicate and object> triples. The concepts in each triple (subject and object) correspond to some of the data items that are the value of the attributes of an XML (eXtensible Markup Language) element or even the element content. On the other side, the relations (predicate) linking concepts can be inferred from the XML elements and their structure. So, the implementation of that ingestion module requires the ability to identify in the given input those data items, to extract their value and print them in the output. Similar ability is required to deduce and print out the predicates.

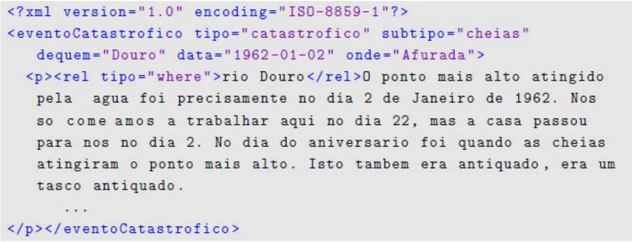
This process can be described using a set of production rules, that will be used, not only as a specification, but also as generative mechanism. Each production rule is a pair: the left hand side is a regular expression (RE) that specifies the element we want to look for; the right hand side is a piece of code that transforms the input data and writes the respective output.

So, our proposal to develop this first stage M1 is to analyze the elements and structure that can appear in the input documents, write a RE-based collection of production rules, and resort to a compiler generator (or a text filter generator) to derive the final program. Notice that the document analysis is a systematic and a formal task because it is supported on the XML document definition (DTD or XML Schema).

Now we describe how this proposal was implemented in the Museum of the Person (MP) case study, based on an ontology composed of the concepts: Person, Event, Place, Date, etc., and relations: participatedIn, carriedOutBy, tookPlace, hasTimeSpan, etc.. In the referred project, we used CIDOC-CRM[1], FOAF[2], and DBpedia[3] nomenclatures, as will be noticed in the examples and pictures along the next sections. For more information on this concrete ontology, please see: *http://npmp.epl.di.uminho.pt/cidoc_foaf_db.html*.

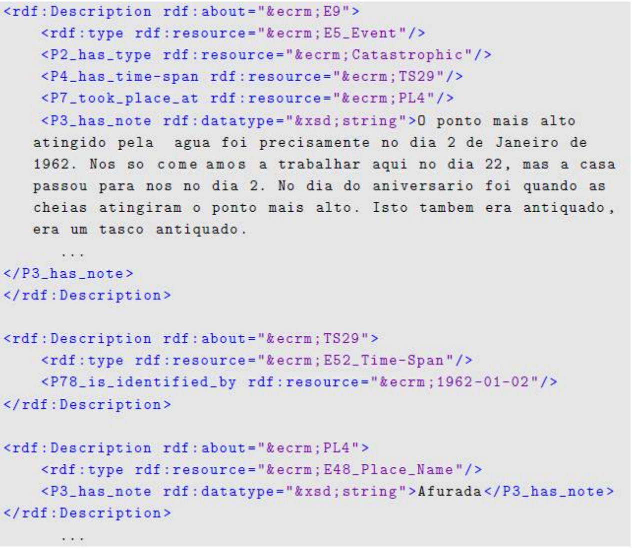
To process the digital repository composed of three types of documents (basic identification, BI; photography Captions*;* and Edited Interview) we have built a tool called XML2RDF translator to perform the ingestion task [M1], that is, to obtain and process input data, automatically, producing a triple store [7] [8] [9].

The text filter was developed using the Compiler Generator system AnTLR[4] (Another Tool for Language Recognition) integrated in *AnTLRWorks* tool, version 2.1 [7] [9]. AnTLR generates a lexical analyzer that implements the desired text filter for data extraction, based on a set of regular expressions. This text filter receives as input an XML (eXtensible Markup Language) document, like the one presented in Figure 2. After analyzing and processing it, the translator will output an RDF (Resource Description Framework) description, as shown in Figure 3.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<eventoCatastrofico tipo="catastrofico" subtipo="cheias"
   dequem="Douro" data="1962-01-02" onde="Afurada">
  <p><rel tipo="where">rio Douro</rel>O ponto mais alto atingido
   pela  agua foi precisamente no dia 2 de Janeiro de 1962. Nos
   so come amos a trabalhar aqui no dia 22, mas a casa passou
   para nos no dia 2. No dia do aniversario foi quando as cheias
   atingiram o ponto mais alto. Isto tambem era antiquado, era um
   tasco antiquado.
      ...
</p></eventoCatastrofico>
```

Figure 2. An XML input document



```
<rdf:Description rdf:about="&ecrm;E9">
    <rdf:type rdf:resource="&ecrm;E5_Event"/>
    <P2_has_type rdf:resource="&ecrm;Catastrophic"/>
    <P4_has_time-span rdf:resource="&ecrm;TS29"/>
    <P7_took_place_at rdf:resource="&ecrm;PL4"/>
    <P3_has_note rdf:datatype="&xsd;string">O ponto mais alto
   atingido pela  agua foi precisamente no dia 2 de Janeiro de
   1962. Nos so come amos a trabalhar aqui no dia 22, mas a casa
   passou para nos no dia 2. No dia do aniversario foi quando as
   cheias atingiram o ponto mais alto. Isto tambem era antiquado,
   era um tasco antiquado.
      ...
</P3_has_note>
</rdf:Description>

<rdf:Description rdf:about="&ecrm;TS29">
    <rdf:type rdf:resource="&ecrm;E52_Time-Span"/>
    <P78_is_identified_by rdf:resource="&ecrm;1962-01-02"/>
</rdf:Description>

<rdf:Description rdf:about="&ecrm;PL4">
    <rdf:type rdf:resource="&ecrm;E48_Place_Name"/>
    <P3_has_note rdf:datatype="&xsd;string">Afurada</P3_has_note>
</rdf:Description>
      ...
```

Figure 3. An RDF output document

The transformation steps which make possible to transform the input file (Figure 2) into the output file (Figure 3) will be explained in detail below. The actual implementation of the XML2RDF functionality, described above, is split into three files, namely [7]: *XML2RDF.g4*, an ANTLR lexer grammar, organized in 'modes', that contains the set of production rules (RE-pattern / reaction) that filters the input files; *Person.java*, a Java class that defines the internal representation for the information we need to extract and process concerning a person

[1] In: *http://www.cidoc-crm.org/*. Accessed: 2016-03-16.

[2] In: *http://www.foaf-project.org/*. Accessed: 2016-03-18.

[3] In: *http://wiki.dbpedia.org/*. Accessed: 2016-03-18.

[4] In: *http://www.antlr.org/*. Accessed: 2016-09-14.

(an interviewee); *MainLexerXML2RDF.java* the main program that orchestrates the other modules to execute their tasks in order to implement the Translator.

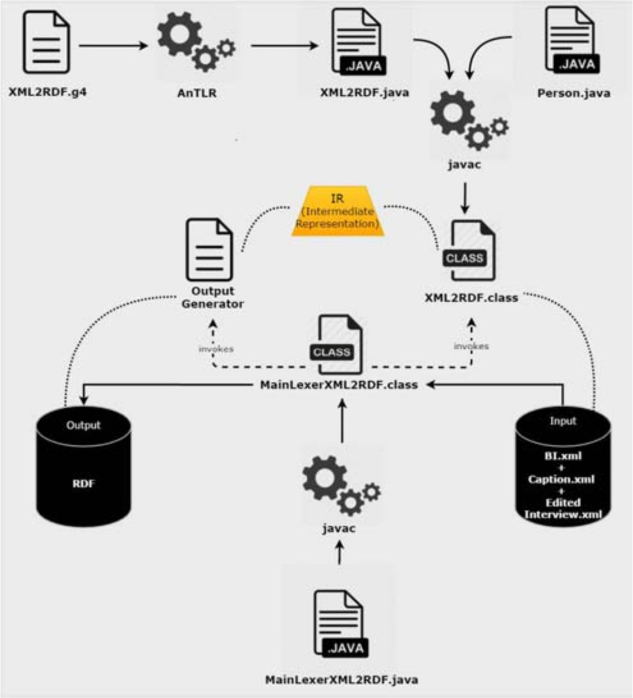
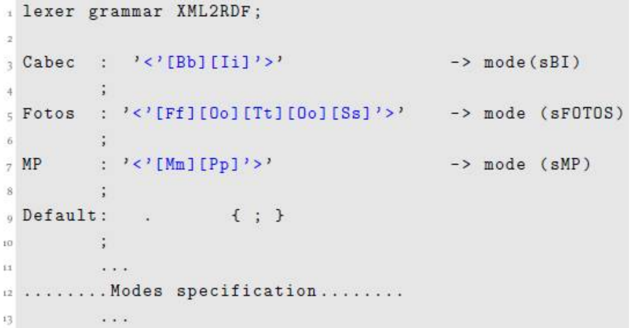


Figure 4. Architecture of Ingestion Function [XML2RDF]

Figure 4 depicts the architecture of Ingestion Function [XML2RDF], the data extractor and *RDF Generator*, based on those three files. Through the *XML2RDF.g4* grammar file, ANTLR generates the *XML2RDF.java* class that is compiled, including the *Person.java* class, to create the desired *XML2RDF* processor [7].

The automatic translation is specified by an *AnTLR Lexer grammar*, shown in Figure 5. This figure shows three transformation rules to process the beginning of the global specification. These three rules (*Cabec*, *Fotos* and *MP*) correspond to the three input files (*BI*, *Photo Captions* and *edited Interview*), respectively [7] [9].



```
lexer grammar XML2RDF;

Cabec   :  '<'[Bb][Ii]'>'               -> mode(sBI)
        ;
Fotos   : '<'[Ff][Oo][Tt][Oo][Ss]'>'    -> mode (sFOTOS)
        ;
MP      : '<'[Mm][Pp]'>'                -> mode (sMP)
        ;
Default:    .       { ; }
        ;
        ...
........Modes specification........
        ...
```

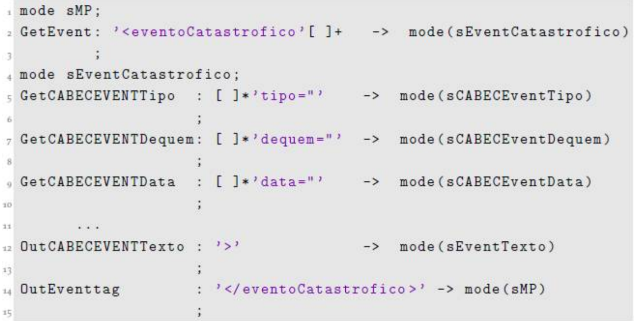
Figure 5. XML2RDF Lexer Grammar in AnTLR notation

A rule contains a name and a pair consisting of a Regular Expression and a Semantic Action written in Java. Regular Expression defines the text pattern to be found in the entry, and Semantic Action specifies how the found concrete text will be transformed [7] [9].

Thus, when the extractor reads an XML tag that determines the start of one of the three input files, it enters a special AnTLR mode to process the contents of that document [7] [9].

To better explain the AnTLR modes, Figure 6 shows an excerpt from the main mode. This excerpt processes the *Catastrophic Event*, when narrated by the person.

In this case, the extractor when it finds the block opening mark, which corresponds to the *Catastrophic Event*, activates the appropriate mode to process the contents of the block. When finding the block closing mark, the processor exits the mode and returns to the initial mode.

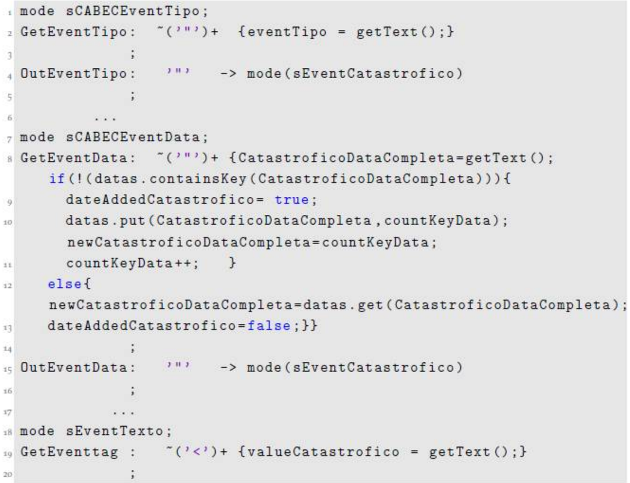


```
1  mode sMP;
2  GetEvent: '<eventoCatastrofico'[ ]+    ->  mode(sEventCatastrofico)
3          ;
4  mode sEventCatastrofico;
5  GetCABECEVENTTipo   : [ ]*'tipo="'    ->  mode(sCABECEventTipo)
6                      ;
7  GetCABECEVENTDequem: [ ]*'dequem="'  ->  mode(sCABECEventDequem)
8                      ;
9  GetCABECEVENTData   : [ ]*'data="'    ->  mode(sCABECEventData)
10                     ;
11        ...
12 OutCABECEVENTTexto : '>'               ->  mode(sEventTexto)
13                     ;
14 OutEventtag         : '</eventoCatastrofico>' -> mode(sMP)
15                     ;
```

Figure 6. Lexer Grammar: Mode to cope with 'Events' in an interview

The three initial auxiliary modes (see lines 5-10) contain specific rules for extracting information from tag attributes. The fourth auxiliary mode (see lines 12-13) contains specific rules for extracting the description of the *Catastrophic Event.*

The rules executed (the modes activated at line 5-13 of Figure 6) to analyze and extract information from XML documents repository are presented in Figure 7.

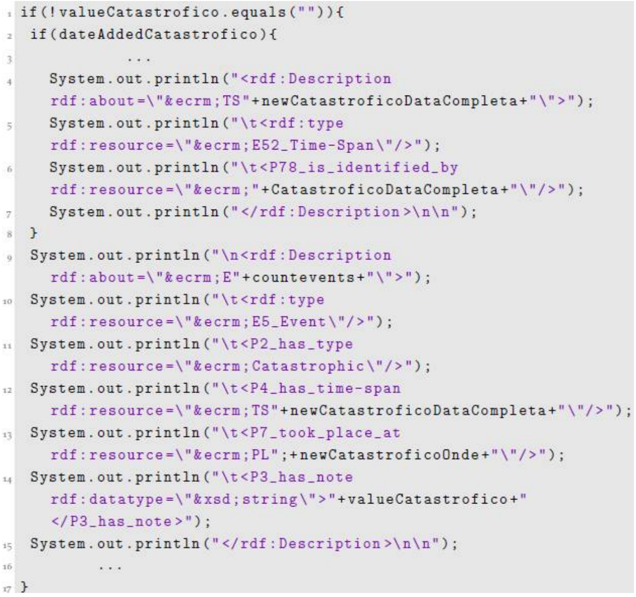


```
1  mode sCABECEventTipo;
2  GetEventTipo:  ~('"')+  {eventTipo = getText();}
3                ;
4  OutEventTipo:   '"'   -> mode(sEventCatastrofico)
5                ;
6            ...
7  mode sCABECEventData;
8  GetEventData:  ~('"')+ {CatastroficoDataCompleta=getText();
     if(!(datas.containsKey(CatastroficoDataCompleta))){
9      dateAddedCatastrofico= true;
10     datas.put(CatastroficoDataCompleta,countKeyData);
       newCatastroficoDataCompleta=countKeyData;
11     countKeyData++;   }
12   else{
     newCatastroficoDataCompleta=datas.get(CatastroficoDataCompleta);
13   dateAddedCatastrofico=false;}}
14               ;
15 OutEventData:   '"'   -> mode(sEventCatastrofico)
16               ;
17             ...
18 mode sEventTexto;
19 GetEventtag :   ~('<')+ {valueCatastrofico = getText();}
20               ;
```

Figure 7. Lexer Grammar: auxiliary Modes

The code block between lines 2 and 15 (Figure 7), has the function of extracting information from the attributes of the tag. In this case, the *type of event* (line 1-5) and the *date* (lines 7-16) are represented. Lines 18-20 of the code block are intended to extract the description of the event.

The grammar fragment responsible for the generation of the RDF output file is presented in Figure 8. The code block between lines 2 and 8, is to create the RDF of the *event date*. Lines 9 and 15, are intended to create the *Catastrophic Event* RDF, which includes the *event type*, *date*, *local*, and *event description*.



```
1  if(!valueCatastrofico.equals("")){
2   if(dateAddedCatastrofico){
3            ...
4     System.out.println("<rdf:Description
      rdf:about=\"&ecrm;TS"+newCatastroficoDataCompleta+"\">");
5     System.out.println("\t<rdf:type
      rdf:resource=\"&ecrm;E52_Time-Span\"/>");
6     System.out.println("\t<P78_is_identified_by
      rdf:resource=\"&ecrm;"+CatastroficoDataCompleta+"\"/>");
7     System.out.println("</rdf:Description>\n\n");
8   }
9   System.out.println("\n<rdf:Description
      rdf:about=\"&ecrm;E"+countevents+"\">");
10  System.out.println("\t<rdf:type
      rdf:resource=\"&ecrm;E5_Event\"/>");
11  System.out.println("\t<P2_has_type
      rdf:resource=\"&ecrm;Catastrophic\"/>");
12  System.out.println("\t<P4_has_time-span
      rdf:resource=\"&ecrm;TS"+newCatastroficoDataCompleta+"\"/>");
13  System.out.println("\t<P7_took_place_at
      rdf:resource=\"&ecrm;PL";+newCatastroficoOnde+"\"/>");
14  System.out.println("\t<P3_has_note
      rdf:datatype=\"&xsd;string\">"+valueCatastrofico+"
      </P3_has_note>");
15  System.out.println("</rdf:Description>\n\n");
16          ...
17 }
```

Figure 8. Lexer Grammar: Print Mode

This grammar fragment is composed by the rules executed at the end of the processing to print out the RDF triples built in the internal representation.

In the next section, we will detail the construction of Virtual Learning Spaces (VLS) to display in a Web browser the information extracted by the XML2RDF translator.

III. Creating Learning Spaces on the web

According to Schweibenz, *Virtual Learning Spaces are virtual spaces that offer different points of access to their virtual visitors. The information is presented in a manner geared to the context rather than being oriented to objects. In addition, the virtual space providing diversified linked information in an attractive interface captivates easily the attention of the visitor and can be seen as a teacher motivating him to learn a specific topic—in this sense, that space can be thought as a Virtual Learning Space* [1].

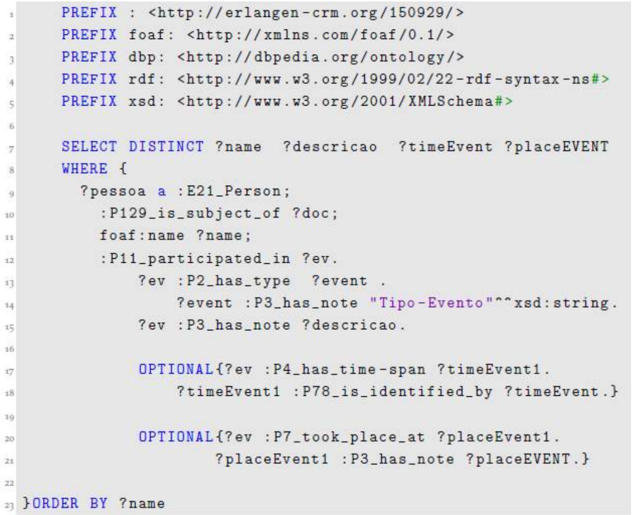
In the context of our research, we aim at exploring efficient and effective ways to create Virtual Learning Spaces (VLS) to display the information (gathered by module M1, responsible for the Ingestion function) in a manner that allows a free conceptual navigation, with an attractive interface that easily captures the visitor's attention and helps him in acquiring knowledge in the museum's domain. The approach we propose is driven by a domain ontology (the one populated in the first stage). On one hand that ontology will determine how to query the data store; on the other hand, the ontology defines how to expose the museum objects in such a way that the visitor can flow from one to another according to the concepts they represent and the ontological relations linking them.

Once again, we illustrate our generic proposal, reporting our experience in the development of Museum of the Person (MP) case study; design and implementation decisions are also discussed.

To display in the Virtual Learning Spaces the information stored in the TripleStore (in our case study we used the Triple Store Database, called Apache Jena TDB[5]) we create a VLS generator to send queries and process the returned data, thus generating the Virtual Learning Spaces.

The VLS Generator consists of two parts: *SPARQL Endpoint* that receives and interprets SPARQL (SPARQL Protocol and RDF Query Language) queries, accesses the TripleStore and returns the answers (in our case, SPARQL endpoint used was Apache Jena Fuseki[6]); and *Query Processor* that generates SPARQL queries according to the requirements of the showroom, sends them to the SPARQL Endpoint and after receiving the response, combines the data returned to configure the VLS [6] [7] [8] [9].

Some queries were created to verify if the information wanted coincided with the one returned. Interviewee and his events is an example of one of the SPARQL queries created, and it is presented in Figure 9.



```
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#>
5      PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
6
7      SELECT DISTINCT ?name  ?descricao  ?timeEvent ?placeEVENT
8      WHERE {
9        ?pessoa a :E21_Person;
10         :P129_is_subject_of ?doc;
11         foaf:name ?name;
12         :P11_participated_in ?ev.
13             ?ev :P2_has_type  ?event .
14                 ?event :P3_has_note "Tipo-Evento"^^xsd:string.
15             ?ev :P3_has_note ?descricao.
16
17             OPTIONAL{?ev :P4_has_time-span ?timeEvent1.
18                 ?timeEvent1 :P78_is_identified_by ?timeEvent.}
19
20             OPTIONAL{?ev :P7_took_place_at ?placeEvent1.
21                     ?placeEvent1 :P3_has_note ?placeEVENT.}
22
23 }ORDER BY ?name
```

Figure 9. SPARQL Query: Interviewee and his events

In the query of Figure 9, the code block between lines 9 and 15 has the function to search for all respondents *(E21 Person)* that participated in events *(:P11_participated_in)*. These events can be of several types *(:P2_has_type)*, such as wedding,

[5] In: *https://jena.apache.org/documentation/tdb/index.html*. Accessed: 2016-06-01.

[6] In: *https://jena.apache.org/documentation/fuseki2/index.html*. Accessed: 2016-06-01.

catastrophic, political, among others. As there are events that do not have the date *(:P4_has_time-span)* and the local *(:P7_took_place_at)* properties, simultaneously, we decide in the OPTIONAL argument to properly manage this inconsistency (lines 16 - 17 and lines 20 - 21, respectively).

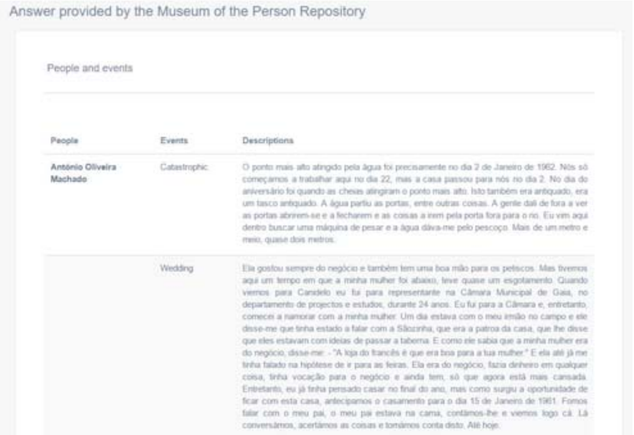
Reusing the SPARQL queries manually created, that script selects those needed for a concrete request and sends the queries to the SPARQL endpoint. It was created a Python script that reuses the SPARQL queries manually created. That script selects those needed for a concrete request and sends them to SPARQL Endpoint and after receiving the answer, combines the data returned to configure the Virtual Learning Spaces (VLS). To create and format web pages, this Python script, uses HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) [7] [8] [9].

This Python script uses the *SPARQLWrapper* library to query the SPARQL endpoint about all *Interviewee and his events* (query shown in Figure 9). It requests the results in a JSON (JavaScript Object Notation) format so that it can easily iterate through the returned data [7].

After preparing the SPARQL query to send off, the script sends the query within a *try/except* block so that it can check for communication problems before attempting to render the results [7]. As is a CGI (Common Gateway Interface) script, it creates HTML and sends it to a browser, a *Content-type* header for sending the actual web page. The script also includes Python code to return the results of the query [7].

An example of the final output of the Python script can be seen in Figure 10. This figure displays a response to the query: *Interviewee and his events.*

Finally, a form was created to execute a query presented in the Python script (Figure 9) and to obtain as a response the web page that lists the *Interviewees and their events* (Figure 10). Notice that, each Virtual Learning Space is built following the template of the web page with data extracted from the data storage [7] [8] [9].



Answer provided by the Museum of the Person Repository

People and events

People	Events	Descriptions
António Oliveira Machado	Catastrophic	O ponto mais alto atingido pela água foi precisamente no dia 2 de Janeiro de 1962. Nós só começamos a trabalhar aqui no dia 22, mas a casa passou para nós no dia 2. No dia do aniversário foi quando as cheias atingiram o ponto mais alto. Isto também era antiquado, era um tasco antiquado. A água partiu as portas, entre outras coisas. A gente dali de fora a ver as portas abrirem-se e a fecharem e as coisas a irem pela porta fora para o rio. Eu vim aqui dentro buscar uma máquina de pesar e a água dava-me pelo pescoço. Mais de um metro e meio, quase dois metros.
	Wedding	Ela gostou sempre do negócio e também tem uma boa mão para os petiscos. Mas tivemos aqui um tempo em que a minha mulher foi abaixo, teve quase um esgotamento. Quando viemos para Candelo eu fui para representante na Câmara Municipal de Gaia, no departamento de projectos e estudos, durante 24 anos. Eu fui para a Câmara e, entretanto, comecei a namorar com a minha mulher. Um dia estava com o meu irmão no campo e ele disse-me que tinha estado a falar com a Sãozinha, que era a patroa da casa, que lhe disse que eles estavam com ideias de passar a taberna. E como ele sabia que a minha mulher era do negócio, disse-me: - "A loja do francês é que era boa para a tua mulher." E ela até já me tinha falado na hipótese de ir para as feiras. Ela era do negócio, fazia dinheiro em qualquer coisa, tinha vocação para o negócio e ainda tem, só que agora está mais cansada. Entretanto, eu já tinha pensado casar no final do ano, mas como surgiu a oportunidade de ficar com esta casa, antecipamos o casamento para o dia 15 de Janeiro de 1961. Fomos falar com o meu pai, o meu pai estava na cama, contámos-lhe e viemos logo cá. Lá conversámos, acertámos as coisas e tomámos conta disto. Até hoje.

Figure 10. Response to the SPARQL query: Interviewee and his events

In the following section, will be presented an example of a guided tour to the Museum of the Person. On this visit, it is possible to see other examples the results and executed queries.

IV. Guided visit to Museum of the Person

To illustrate the outcome of this project, we will detail an example of guided visit through the virtual Museum of the Person. This visit comprises the following steps [7]:

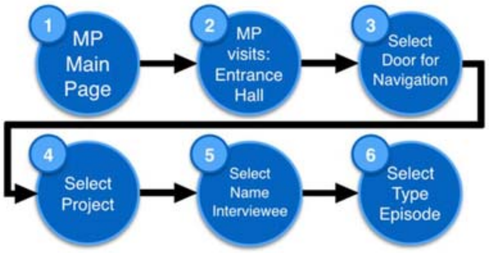
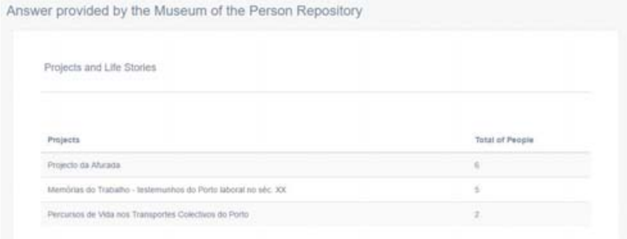


Figure 11.Workflow that guides the Visit to Museum of the Person

The visit starts at: *http://npmp.epl.di.uminho.pt*. At this address the main page of the Museum of the Person will be seen – step (1) [7].

At this point, selecting the *MP: Visits* option in the menu on the left hand side and then the option *Entrance Hall*, we go to the museum's *Entrance Hall* – step (2) [7]. As previously stated, this page contains multiple entrance doors to start the navigation.

In step (3) we will visit the room *Projects and Life Stories* (Figure 12) [7].



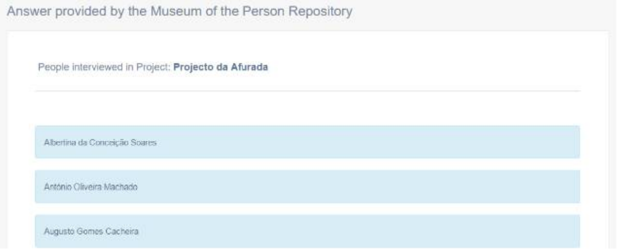
Answer provided by the Museum of the Person Repository

Projects and Life Stories

Projects	Total of People
Projecto da Afurada	6
Memórias do Trabalho - testemunhos do Porto laboral no séc. XX	5
Percursos de Vida nos Transportes Colectivos do Porto	2

Figure 12. Room: Projects and Life Stories

The room *Projects and Life Stories* displays the list of all projects in the repository and the number of life stories per project (Figure 12). The step (4) of this visit is to select the project from that list. In this case, we will visit *Projecto Afurada.* Clicking on that item, *Projecto Afurada*, we enter the room of this project (Figure 13) [7].



Answer provided by the Museum of the Person Repository

People interviewed in Project: **Projecto da Afurada**

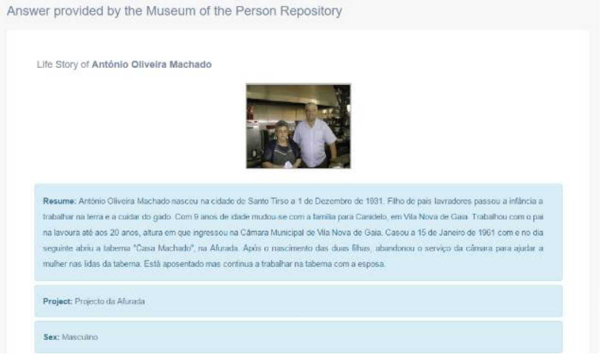
Albertina da Conceição Soares

António Oliveira Machado

Augusto Gomes Cacheira

Figure 13. Room: Projecto Afurada

The room Projecto Afurada lists the name of all the people interviewed in the context of that project. To read the life story of someone we just click on the person's name – step (5). The person chosen in this visit is António Oliveira Machado (Figure 14) [7].



Answer provided by the Museum of the Person Repository

Life Story of **António Oliveira Machado**

Resume: António Oliveira Machado nasceu na cidade de Santo Tirso a 1 de Dezembro de 1931. Filho de pais lavradores passou a infância a trabalhar na terra e a cuidar do gado. Com 9 anos de idade mudou-se com a família para Canidelo, em Vila Nova de Gaia. Trabalhou com o pai na lavoura até aos 20 anos, altura em que ingressou na Câmara Municipal de Vila Nova de Gaia. Casou a 15 de Janeiro de 1961 com e no dia seguinte abriu a taberna "Casa Machado", na Afurada. Após o nascimento das duas filhas, abandonou o serviço da câmara para ajudar a mulher nas lidas da taberna. Está aposentado mas continua a trabalhar na taberna com a esposa.

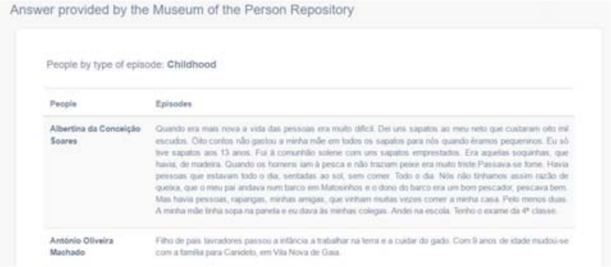
Project: Projecto da Afurada

Sex: Masculino

Figure 14. Life Story of Interviewee: António Oliveira Machado

In this room the life story of António Oliveira Machado is told. Details, such as date and place of birth, profession, qualifications, ancestry, photos, events in which he participated and episodes narrated, can be found and read. There are several types of episodes, such as educational, religious, general, etc.

In that room, where the life story is displayed, besides being possible to read all the episodes narrated by António Oliveira Machado, it is also possible to see their types. Choosing a type (by clicking over the type name) the visitor will learn more about the episodes of that type narrated by all the interviewees. In concrete, we will choose the type *Childhood* and will navigate to a new room that displays all the episodes of that kind (*Childhood*) for all the people interviewed (Figure 15) – step (6). This room allows to relate people and stories regarding the episodes they lived [7].



Answer provided by the Museum of the Person Repository

People by type of episode: **Childhood**

People	Episodes
Albertina da Conceição Soares	Quando era mais nova a vida das pessoas era muito difícil. Dei uns sapatos ao meu neto que custaram oito mil escudos. Oito contos não gastou a minha mãe em todos os sapatos para nós quando éramos pequeninos. Eu só tive sapatos aos 13 anos. Fui à comunhão solene com uns sapatos emprestados. Era aquelas soquinhas, que havia, de madeira. Quando os homens iam à pesca e não traziam peixe era muito triste.Passava-se fome. Havia pessoas que estavam todo o dia, sentadas ao sol, sem comer. Todo o dia. Nós não tínhamos assim razão de queixa, que o meu pai andava num barco em Matosinhos e o dono do barco era um bom pescador, pescava bem. Mas havia pessoas, raparigas, minhas amigas, que vinham muitas vezes comer a minha casa. Pelo menos duas. A minha mãe tinha sopa na panela e eu dava às minhas colegas. Andei na escola. Tenho o exame da 4ª classe.
António Oliveira Machado	Filho de pais lavradores passou a infância a trabalhar na terra e a cuidar do gado. Com 9 anos de idade mudou-se com a família para Canidelo, em Vila Nova de Gaia.

Figure 15. Room: Interviewees by type of episode (Childhood)

V. CONCLUSION

Memory institutions---like archives, libraries or museums---detain, nowadays, many information repositories in the format of natural language digital documents, yet annotated or that can be annotated.

The existence of so many knowledge sources, lets us think on the possibility of organizing such a rich cultural heritage in a conceptual manner that can be disseminated on the Web. These sites, where the physical or abstract objects are exhibited linked among them by well-defined logical relations creating a semantic network (or conceptual map), are called virtual learning spaces (VLS) because they allow the cyber-navigator to learn about a specific domain. We argued along the paper that those VLS resemble exhibition rooms in a traditional museum, and we discussed a systematic way to build such spaces querying a data-store organized as an ontology for the referred domain. Moreover, we also proposed an approach to extract data from the given digital documents repository building automatically the required data-store. Using the Museum of the Person as a case-study, we illustrate how elements (or data item) can be found in the annotated documents to instantiate the ontology triples with concepts (the triple's subject and object) and relations (the triple's predicate). Although the triple storage is the most natural way to store the instances, our population method does not change if a relational database is used as archival (for a detailed and interesting comparison about both approaches see [10]).

A guided tour to the Museum of the Person was included in the paper to prove the feasibility of our proposals. To consolidate the work here presented, it is crucial to apply in the near future the approach to other case-studies. Also, experiments to assess the usability and quality of the VLS's created by our system, must be designed and conducted.

ACKNOWLEDGMENT

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.
The work of Ricardo Martini is supported by CNPq, grant 201772/2014-0.

REFERENCES

[1] W. Schweibenz, "The development of virtual museums, In Virtual Museums, 3rd ed., vol. 57(3), ICOM, 2004.

[2] J. J. Almeida, J. G. Rocha, P. R. Henriques, S. Moreira, and A. Simões, "Museu da Pessoa – arquitetura," in Encontro nacional da associação de bibliotecários, arquivista e documentalistas, ABAD'01. BAD, 2001, URL http://repositorium.sdum.uminho.pt/handle/1822/585.

[3] A. Simões and J. J. Almeida, "Histórias de vida + processamento estrutural = Museu da Pessoa," in XATA 2003 — XML: Aplicações e Tecnologias Associadas, page 16, Braga, Portugal, 2003, UM, URL http://repositorium.sdum.uminho.pt/handle/1822/629.

[4] R. G. Martini, C. Araújo, J. J. Almeida, and P. R. Henriques, "OntoMP, An Ontology to Build the Museum of the Person", New Advances in Information Systems and Technologies, vol. 2, p.p 653–661. Springer International Publishing, Cham, 2016

[5] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing". In International Journal of Human-Computer Studies, pp. 907–928. Kluwer Academic Publishers, 1993.

[6] C. Araújo, R. G. Martini, P. R. Henriques, and J. J. Almeida, "Architectural Approaches to build The Museum of the Person," in Á. Rocha, L. P. Reis, M. P. Cota, O. S. Suárez, and R. Gonçalves, editors, Sistemas y Tecnologías de Información — Atas da 11ª Conferência Ibérica de Sistemas e Tecnologias de Informação, volume vol. 1 – Artículos de la Conferencia, pp. 383–388. AISTI–Associação Ibérica de Sistemas e Tecnologias de Informação, June 2016.

[7] C. Araújo, "Building the Museum of the Person Based on a combined CIDOC-CRM/ FOAF/ DBpedia Ontology", MSc Thesis, Universidade do Minho, 2016.

[8] C. Araújo, R. G. Martini, P. R. Henriques, and J. J. Almeida, "Building the Museum of the Person from RDF Triples and SPARQL", in Communications and Innovations Gazette (ComInG), 2nd ed., vol 1, pp. 1–14, 2016.

[9] C. Araújo, R. G. Martini, P. R. Henriques, and J. J. Almeida, "Annotated documents and expanded CIDOC-CRM Ontology in the automatic Construction of a Virtual Museum", unpublished.

[10] C. M.-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: are they so different? A comparison", Artif. Intell, 4th ed, vol. 38, pp. 271-290, 2012