Digital Collection Creator, Visualizer and Explorer

Luís F. Martins[1]
Algoritmi-DI, University of Minho, Portugal
lmfmrt@gmail.com

Cristiana Araújo
Algoritmi-DI, University of Minho, Portugal
decristianaaraujo@hotmail.com

Pedro Rangel Henriques
Algoritmi-DI, University of Minho, Portugal
pedrorangelhenriques@gmail.com

Abstract

In this paper we introduce and discuss a recent project, called CortaColaEspia, aimed at extending with some extra relevant features the 'Ontology-based Collection Processor' developed previously in the context of a Compilers course. The basic processor, based on the OntoDL tool, was able to read the ontological description of a small collection of objects (cards, pencils, toys, etc.) and produce automatically a web-based exhibition space to display the objects, providing a conceptual navigation through them. The extension under discussion is intended to create a new DSL to describe the details of the exhibition room organization (what concepts and relations to show; where and how to show them; etc.). A second objective consists of a new module to merge two collections, or to enrich a collection with extra information about the collected objects. The last requirement is the incorporation of a natural language processor to analyze the objects' captions or short inscriptions in order to extract information that can create knowledge about a specific domain, a society or an epoch.

2012 ACM Subject Classification Information systems → Ontologies; Applied computing → Extensible Markup Language (XML); Software and its engineering → Domain specific languages

Keywords and phrases Digital Collections, Ontology, DSL, Program Generation

Digital Object Identifier 10.4230/OASIcs.SLATE.2019.15

Funding This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.

1 Introduction

Collecting objects is a very old human habit and the digitalization of collections to manage them on the computer and exhibit the objects on the Web is a very common practice nowadays.

Describing the objects of a museum collection and defining all the exhibition details of those objects, allowing them to be linked conceptually in order to tell stories and transmit knowledge to the space visitors, is a demanding task requiring time and effort but chiefly the know-how of a museum's expert like a curator.

We believe that the exhibition of a small collection gathered by an individual, not a professional, is valuable to the community. Handling these small collections can be useful in different contexts: exhibition of familiar collections in personal web pages; creation of small learning spaces to be used in educational and recreative activities; construction of specific catalogues to be used in e-commerce activities; and so on …

[1] Corresponding author

8th Symposium on Languages, Applications and Technologies (SLATE 2019).
Editors: Ricardo Rodrigues, Jan Janoušek, Luís Ferreira, Luísa Coheur, Fernando Batista, and Hugo Gonçalo Oliveira; Article No. 15; pp. 15:1–15:8
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this context, the paper presents a light alternative for the web publishing and navigation of small collections resorting to Domain-Specific Languages (DSL) as a twofold artifact: a specification language for the actual collection, and the definition of the representation itself [1]. To describe our approach, we first present a DSL, OntoDL, to define collections (car miniatures, toys, pencils, sugar bags, coins, post stamps, etc.) based on ontologies. The ontology specifies the collection's knowledge domain creating a set of concepts (to classify domain objects), and defining relationships (hierarchical or not) among the main concepts (or classes); then individuals (or concrete objects) are linked to concepts using an 'instance-of' special relation. This DSL allows to describe the conceptual level of the collection and the collection objects that will populate the ontology in a very easy way using a light notation [2]. The proposed approach uses a second DSL, CollExhibDL, to describe the exhibition space, resorting again to the ontology concepts and relations, and never concerned with the instances. After discussing the role of the ontology and the related DSLs to define the collection and how to show it, we propose a system that automatize both phases, the collection upload (or the information ingestion) and its exhibition on a web site (information dissemination) [9].

This system will be built based on: the attribute grammars that specify the referred DSLs; and on the compiler generator ANTLR [8] that analyzes the grammars and generates automatically the Java code that implements the desired processors.

The present section (Section 1) serves as an introduction to the paper. Here we present the work context and project motivation, provide a brief description of the proposed system, and then we introduce the case studies used in order to test CortaColaEspia and prove its feasibility and versatility. This paper covers two of the modules of the system we intend to build. A short presentation of the already developed OntoDL language and processor is the content of Section 2. The specific language CollExhibDL and the Web exhibition rooms generated from the ontology and the room description are discussed in Section 4. A detailed overview of the system's architecture is presented in Section 3; this section explains the system modules and the connections among them. Before the conclusion, in Section 6, another section (Section 5) discusses related work.

1.1 Case Studies: Introduction

In order to prove the usefulness of our system, three case studies were carefully chosen:

Religious Greeting Cards. The first case study was the major case for the development of the OntoDL tool already built. The collection consists on greeting cards given to attendants of religious events, such as holy communions and weddings, that usually contain information about the event and where it took place, the person and saint involved, and a handwritten message.

Sugar Packets. Some brands of coffee make limited edition collections of sugar packets with curiosities about a specific event or about a region and its uses or history. Usually each element of these collections has a small description of an individual of the subject matter and a small graphic/picture representing the topic described.

Tickets. The last case study is a collection of tickets for sport events or music concerts that contain useful information to prove this project concept, like locations, company advertising and clubs or artists.

2 An Ontology for Digital Collections

In this section we try to explain the first developed module of this system, the OntoDL processor and our perspective in answering this module objectives. In this section we first briefly introduce the definition of Ontology and then we present the formal language, OntoDL, to describe, in a natural and syntectic way, ontologies at a abstract level (just concepts and relations) or at a concrete level adding indicuals to instantiate classes. In a third subsection, we will use OntoDL language to specify the ontologies for the three collections chosen as case-studies.

2.1 Ontologies

An ontology may be described as a body of formally represented knowledge, based on *conceptualization*, making possible to represent an abstract simplified view of the world according to the wished purpose [2].

The reasons to develop an ontology may be summarized in five items:

- To share common understanding of the structure of information among people or software agents;
- To enable the reutilization of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyze domain knowledge.

2.2 Case Study: OntoDL Description

OntoDL is a DSL, developed by our group, that allows for a fast and easy specification of the structure of any given ontology and also describe its population.

Listing 1 Example of an Ontology in OntoDL.

```
Ontology Packets

concepts {
    Packet[title:string,imagePath:string,
           packetNumber:int,description:string],
    Brand[name:string],
    Collection[name:string, series_size:int, year:int],
    Event[name:string, category:string]
    }
individuals {
    c1, p11, chris
    }
relations {
    belongs, madeBy, reference
    }
triples {
    Packet = belongs => Collection;
    Packet = madeBy => Brand;
    Collection = reference => Event;
    Collection = belongs => Brand;
    chris =iof=> Brand[name='Christina'];
    c1 =iof=> Collection[name='Expressoes Tipicas do Norte',
        series_size='9'];
23        c1 =belongs=> chris;
24        p11 =iof=> Packet[title='Olha estes dois a picar o ponto',
25            imagePath = 'expressoes_christina_namorar.png',
26            packetNumber='3', description='Significado: Olha estes
                  dois a namorar'];
27        p11 =madeBy=> chris;
28        p11 =belongs=> c1  }
```

As shown in Listing 1, an ontology is described starting in the first line with its name. In the following lines, 3 to 15, the concepts, the individuals and the relations that characterize the ontology are declared. Please notice that in the first block 'concepts' attributes (defined by a pair name and type) can be assigned to concepts under definition. In the last block, lines 16 to 28, the triples (linking concepts and individuals using the relation) are finally specified. The OntoDL file, shown in Listing 1, is then processed by an OntoDL compiler. That compiler, produced automatically by ANTLR from the OntoDL Attribute Grammar, validates all the input specification and generates both a OWL file and a DOT file (to draw the ontology graph).

When processed by our OntoDL tool, we successfully generated a graph representation for each case study. For the sake of space we just included, in Figure 1, the graphic visualization for the second case study. The graphs obtained for all the case studies can be found at `www.di.uminho.pt/~gepl/CORTACOLAESPIA`.

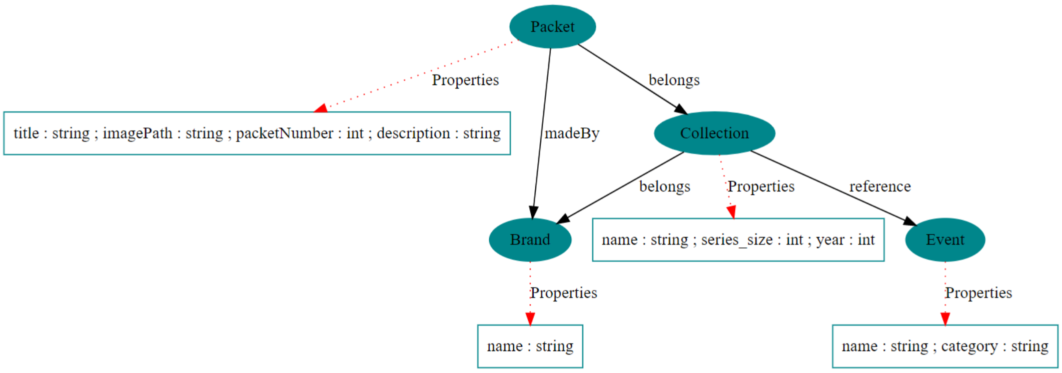

Packet
Properties
belongs
title : string ; imagePath : string ; packetNumber : int ; description : string
madeBy
Collection
belongs
Properties
reference
Brand
name : string ; series_size : int ; year : int
Event
Properties
Properties
name : string
name : string ; category : string

Figure 1 OntoDL Generated Sugar Packets Graph.

3 CortaColaEspia: the Architecture

The system we proposed to develop can be seen has a combination of three main modules, one for the knowledge specification, one for the viewing specification and another for the merging of knowledge.

The architecture of the system is represented by the block diagram shown in Figure 2.

The CPM (in blue) is composed by the OntoDL tool explained above. It generates an OWL file providing an OntoDL collection specification is given. This module consists on a DSL that specificies the structure of an ontology and its population.

The CVNM (in green) allows to generate a XSL Transformation file to be applied in the previously generated OWL, given a CollExhibDL collection visualization file. This module allows the user to specify which fields to show and which to give more emphasis to. Also the user can specify the whole organization of the information to display, such as whether and where it show an index or where the images are shown.

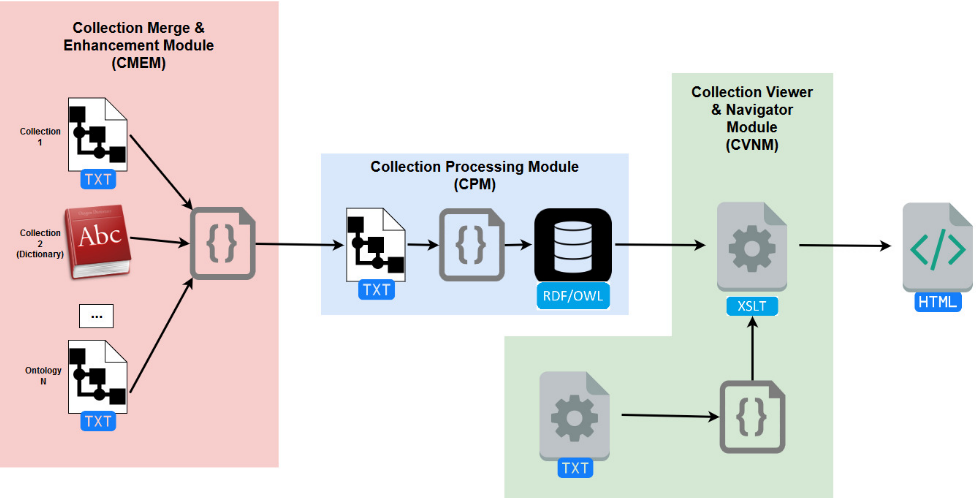

Collection Merge & Enhancement Module (CMEM)
Collection 1
TXT
Collection 2 (Dictionary)
Abc
...
Ontology N
TXT
Collection Processing Module (CPM)
TXT
RDF/OWL
Collection Viewer & Navigator Module (CVNM)
XSLT
TXT
HTML

Figure 2 System Architecture.

The CMEM (in red) allows to merge multiple similar collections and successfully generate a single OntoDL file to be processed as a whole. In the merging module the main objective is to merge data from diverse ontologies, mapping them, in order to complement or expand the existing knowledge.

4 A Website to Explore Collections

This section is splitted into two subsections. The first one is used to introduced a new formal language to describe the layout desired to exhibit a specific collection given its ontological definition, and the second part contains an example of the use of that language to state how the collection (second case-study, specified in the previous section) shall be shown.

4.1 CollExhibDL

This subsection is divided into the DSL Approach that was taken and the Grammar developed to process said DSL.

The components required to achieve a basic template that works for every collection are:

- The concept that we are looking to display;
- Every attribute of that concept that we want to see;
- Every related concept of interest.

Focusing on this three aspects, a second and new DSL was designed aimed at controling the layout and the final details of the Webpage to be created to exhibit the collection.

This DSL is composed of three key elements: a **card** tag, that works as a container or a menu of a respective class and which can be contained in another instance of card; a **show** tag, descendant of card allowing to show attributes of that class; and a **relation** tag, also descendant of card allowing to relate its specific instance with its parent card.

An instance of this DSL is represented in the Listing 2.

Listing 2 Snippet of a basic specification written in CollExhibDL

```
view ’Sugar Packets’ green
card ’#Brand’
   show ’#name’
   card ’#Collection’ relation ’#belongs’
      show ’#name’
      card ’#Packet’ relation ’#belongs’
         show ’#title’ description ’Packet’
         showI ’#imagePath’
         showC ’#packetNumber’ description ’Packet Number’
         showC ’#description’ description ’Description’ . . .
```

Each card is delimited by a punctuation point, allowing to have both sibling and descendant cards.

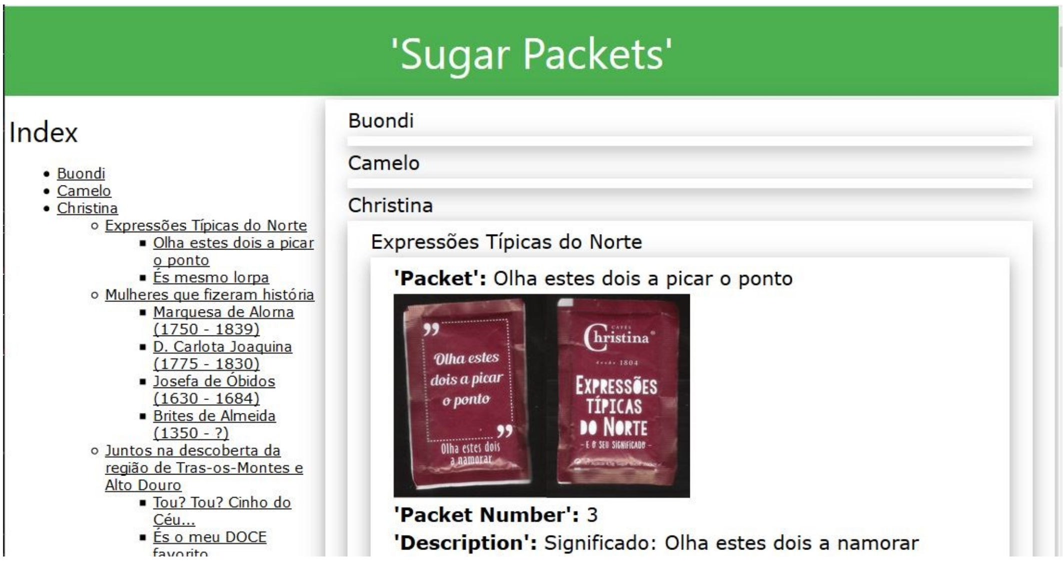
In order to process any given specification written in CollExhibDL, a grammar was developed, defining syntactic rules and semantic validations. This grammar is described in Listing 3.

Listing 3 CollExhibDL Grammar

```
view : ’view’ label=PAL card* ;

card : ’card’ concept=PAL (relation)? (show|showC|showI)* (
    subcard)* ’.’ ;
subcard : card ;

show : ’show’ attr=PAL (’description’ desc=PAL)? ; %% Show
    content in coll. page and index
showC : ’showC’ attr=PAL (’description’ desc=PAL)? ; %% Show
    content in coll. page only
showI : ’showI’ attr=PAL (’description’ desc=PAL)? ; %% Show
    image

relation : ’relation’ rel=PAL ;

PAL: ([a-zA-Z][-_a-zA-Z0-9?]*)|(’\’’ ~[’\’’]* ’\’’)|(’\"’
    ~[’\’’]* ’\"’) ;
NUM: ’-’?[0-9]+ ;
Sep: (’\r’? ’\n’ | ’ ’ | ’\t’)+ -> skip ;
Comment: ’%%’~(’\n’)+ -> skip ;
```

4.2 Case Studies: the Exhibition

When the previously presented ontology representations of the collections are processed by this tool, it successfully generates the HTML websites as expected. In Figure 3 we show one of those websites to illustrate the processor output. Other examples, obtained for the three case studies can be seen at `www.di.uminho.pt/~gepl/CORTACOLAESPIA`.



'Sugar Packets'

Index

- Buondi
- Camelo
- Christina
  - Expressões Típicas do Norte
    - Olha estes dois a picar o ponto
    - És mesmo lorpa
  - Mulheres que fizeram história
    - Marquesa de Alorna (1750 - 1839)
    - D. Carlota Joaquina (1775 - 1830)
    - Josefa de Óbidos (1630 - 1684)
    - Brites de Almeida (1350 - ?)
  - Juntos na descoberta da região de Tras-os-Montes e Alto Douro
    - Tou? Tou? Cinho do Céu...
    - És o meu DOCE favorito

Buondi

Camelo

Christina

Expressões Típicas do Norte

'Packet': Olha estes dois a picar o ponto

Olha estes dois a picar o ponto
Olha estes dois a namorar
Christina
1804
Expressões Típicas do Norte
- E O SEU SIGNIFICADO -

'Packet Number': 3

'Description': Significado: Olha estes dois a namorar

Figure 3 CollExhibDL Generated Sugar Packets HTML.

5 Related Work

Although anyone can find over the Web many sites exhibiting small collections – for instance, Colnect[2] that contains more than 35 instances, Portal do Colecionismo[3], Pacotada[4], or even commercial sites like Casa do Colecionador[5] – as fas as we know, there is not any automatic generation tool that is able to accept a formal description of the collection to build the site. However the project was mainly inspired in CaVa platform, developed by Ricardo Martini [6, 5], to create virtual museums from ontology-based descriptions of the desired museum.

Also, concerning the use of DSL [4, 7] to generate other family of information systems, we can cite [9, 3].

6 Conclusion

All people have hobbies, some like to watch movies, others like to play sports, others like to collect objects... Collectors usually do not only have one collection, but multiple collections of different objects. In this context, we found it important to create a system (CortaColaEspia) that can store and exhibit in Web pages collections of objects and even merge different collections.

So in a first phase we identified 3 different case studies (collections) to store in our system, and we described these 3 collections by means of ontologies using the formal language OntoDL. Following a DSL Approach, we then implemented a grammar to define a new formal language (CollExhibDL) to describe the layout desired to display a specific collection in a Web page. Given the collection ontological definition, that language can be used to declare how each collection shall be shown and navigated.

[2] `https://colnect.com/`, last access in April 2019
[3] `http://www.portaldocolecionismo.pt`, last access in April 2019
[4] `https://www.pacotada.com/`, last access in April 2019
[5] `http://casadocolecionador.net/`, last access in April 2019

As future work it is necessary to develop the module that allows the merge of different collections to get a richer exhibition, and then test the overall system with other collections. We also want to encourage collectors to manage their collection with our system so that we can assess CortaColaEspia with real size cases.

References

1 Ines Ceh, Matej Crepinsek, Tomaz Kosar, and Marjan Mernik. Ontology driven development of domain-specific languages. *Comput. Sci. Inf. Syst.*, 8(2):317–342, 2011.

2 Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human-Computer Studies*, pages 907–928. Kluwer Academic Publishers, 1993.

3 Milan Kosanović, Igor Dejanovic, and Gordana Milosavljevic. Applang – A domain-specific language for rapid development of data-oriented mobile applications. In *Int. Conf. of Numerical Analysis and Applied Mathematics, ICNAAM*, volume 1738, page 240003, June 2016. doi:10.1063/1.4952022.

4 Tomaž Kosar, Sudev Bohra, and Marjan Mernik. Domain-Specific Languages: A Systematic Mapping Study. *Information and Software Technology*, pages –, 2015. Accepted manuscript (article in Press). doi:10.1016/j.infsof.2015.11.001.

5 Ricardo G. Martini, Cristiana Araújo, Pedro Rangel Henriques, and M.João Varanda Pereira. CaVa: an example of the automatic Generation of Virtual Learning Spaces. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo, editors, *Trends and Advances in Information Systems and Technologies, WorldCist2018*, volume 745 of *Advances in Intelligent Systems and Computing*, pages 633–643. Springer International Publishing, Cham, March 2018. doi:10.1007/978-3-319-77703-0_63.

6 Ricardo Giuliani Martini. *Formal Description and Automatic Generation of Learning Spaces based on Ontologies.* PhD thesis, University of Minho, September 2018.

7 Marjan Mernik, Jan Heering, and Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, December 2005. doi:10.1145/1118890.1118892.

8 Terence Parr. ANTLR. http://www.antlr.org/, 2016. Accessed: 2016-09-14.

9 Agnis Stibe and Janis Bicevskis. Web Site Modeling and Prototyping Based on a Domain-Specific Language. In *Computer Science and Information Technologies Vol. 751*, pages 7–21, Riga, Latvia, 2009. Scientific Paper, University of Latvia.