Knowledge Graph Implementation of Archival Descriptions Through CIDOC-CRM

Inês Koch, Nuno Freitas(✉), Cristina Ribeiro, Carla Teixeira Lopes, and João Rocha da Silva

INESC-TEC, Faculty of Engineering, University of Porto, Porto, Portugal
up201403153@letras.up.pt, up201404739@fe.up.pt, mcr@fe.up.pt, ctl@fe.up.pt, joaorosilva@gmail.com

Abstract. Archives have well-established description standards, namely the ISAD(G) and ISAAR(CPF) with a hierarchical structure adapted to the nature of archival assets. However, as archives connect to a growing diversity of data, they aim to make their representations more apt to the so-called linked data cloud. The corresponding move from hierarchical, ISAD-conforming descriptions to graph counterparts requires state-of-the-art technologies, data models and vocabularies. Our approach addresses this problem from two perspectives. The first concerns the data model and description vocabularies, as we adopt and build upon the CIDOC-CRM standard. The second is the choice of technologies to support a knowledge graph, including a graph database and an Object Graph Mapping library. The case study is the Portuguese National Archives, Torre do Tombo, and the overall goal is to build a CIDOC-CRM-compliant system for document description and retrieval, to be used by professionals and the public. The early stages described here include the design of the core data model for archival records represented as the ArchOnto ontology and its embodiment in the ArchGraph knowledge graph. The goal of a semantic archival information system will be pursued in the migration of existing records to the richer representation and the development of applications supported on the graph.

Keywords: Archival description · CIDOC-CRM · Torre do Tombo National Archives · Knowledge graph

1 Introduction

The Portuguese National Archives, Torre do Tombo (ANTT) hold a vast collection of unique cultural objects and provide diverse services to the community, from support to scientific research to the access to legal documents. Given the huge effort invested in the creation of millions of document records, it is essential

This work is financed by National Funds through the Portuguese funding agency, FCT—Fundação para a Ciência e a Tecnologia within project DSAIPA/DS/0023/2018.

A. Doucet et al. (Eds.): TPDL 2019, LNCS 11799, pp. 99–106, 2019.
https://doi.org/10.1007/978-3-030-30760-8_8

to keep them in representations that are interoperable both within archives and across other knowledge sources. The archives are therefore exploring alternative representations to the traditional hierarchical standards [1].

This work is running as part of the EPISA project[1] (Entity and Property Inference for Semantic Archives). The project will define a new data model for the archives and a prototype knowledge graph to support applications for professionals and for the public. The model and the knowledge graph will be tested on a selection of archival records. To populate the graph, the contents of the records will be explored with automatic methods to extract entities and their relationships, enriching the graph with information in the existing records. The prototype knowledge graph and corresponding user interfaces will support the development of a new archival information system, for which the identification of requirements is also ongoing.

We describe our approach to move from hierarchical, ISAD-conforming descriptions to their graph counterparts. The problem is addressed from two perspectives. The first concerns the data model and description vocabularies and the second relates to the construction of an information system based on a knowledge graph.

2 Description Models for Cultural Heritage

To build a new data model for the archives it is essential to analyze the archival description standards in use at the ANTT and also to go into the details of CIDOC-CRM, selected as the foundation for the new model. It is also important to look at cases in which the CIDOC-CRM has been applied, to take into account their experience.

ISAD(G)—General International Standard Archival Description—provides general guidance on the preparation of archival descriptions and is the basis for the existing records [5]. The standard is based on uniform, multi-level description, and assumes the inheritance of descriptor content down the hierarchy. This provides contextual information for items even in case it is infeasible to describe them individually. Although appropriate for archives, ISAD(G) is based on archival concepts and does not favour relations to data from other sources.

The CIDOC-CRM [4] is being developed by the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM) to provide the museum community with good practice and technologies for documentation. The model is based on events and has at its core Temporal Entities, i.e., things that happened in a specific period of time. Only Temporal Entities can be linked to time and have Time Spans. On the other hand, the Objects (Conceptual Object and Physical Thing), Actors/People and Places, rather than being linked to time, are linked via an event—a Temporal Entity. A Place is a physical location, and it can be geographically referenced [6].

The CIDOC-CRM originated in museums and we can find applications mainly in this domain. The Museo del Prado, in Madrid, is an example of a

[1] http://episa.inesctec.pt/.

museum using the ontology in its Knowledge Graph[2]. The main entities of the Prado semantic network (Artwork, Author, Exhibition and Activity) are represented according to the CIDOC-CRM standard. For other kinds of entities not available on the CIDOC-CRM their model adopted concepts from FRBR (Functional Requirements for Bibliographic Records) and other vocabularies widely used in semantic web projects. The British Museum also uses the CIDOC-CRM, asserting that they were the first arts organization in the UK to publish their collection semantically[3].

The CIDOC-CRM has also been applied outside the museum environment, namely to represent archival metadata. These experiments were based on the `EAD` (Encoded Archival Description) representation of ISAD, taking into account the concepts of archival records and their components, as well as the main concepts of the archival description, namely the hierarchical structure and the inheritance of information down the hierarchy of the levels of description [2].

3 The CIDOC-CRM Ontology in Archives

The design of a flexible data model for the archives led to the analysis of the CIDOC-CRM, version 6.2[4]. This has shown that the CIDOC-CRM is powerful, and specific enough for the description of cultural objects in museums, but obviously lacks concepts that are present in the existing archival models and may not be discarded.

Building on the existing representations for the CIDOC-CRM, our model is represented as an ontology and follows the entities and properties of the CIDOC-CRM where applicable. The first challenge with this ontology is its detail and wide scope. The CIDOC-CRM defines a large number of classes and properties that required an in-depth analysis with the help of the existing documentation. Given that we intend to design a model for archives, we used a sample of 14 archival records, from two different sources, DigitArq (the DGLAB record database)[5] and the Guidelines for Archival Description (ODA) [3]. This allowed us to perform a first selection of the classes and properties based on features from actual records.

The second challenge was that most of the available properties are Object Properties, which relate individuals. Given that archive records include many literal fields with specific semantics, using only the available Data Properties proved insufficient to keep the semantics of the base content of archival records, that only requires basic datatypes, without creating artificial objects just to capture those values.

The CIDOC-CRM only defines eight Data Properties (namely `P3 has note`, `P57 has number of parts`, `P79 beginning is qualified by`, `P80 end is qualified by`, `P81 ongoing throughout`, `P82 at some time within`, `P90`

[2] https://www.museodelprado.es/modelo-semantico-digital/modelo-ontologico.

[3] https://www.britishmuseum.org/.

[4] http://new.cidoc-crm.org/Version/version-6.2.

[5] https://digitarq.arquivos.pt/.

has value and P168 place is identified by), so the ontology needs to be complemented to represent the semantics of the descriptors used in our archival records.

We therefore added several sub-properties to property P3 has note, to allow that distinction while retaining interoperability with CIDOC-CRM. Using the properties in Table 1, we keep the semantics associated to the existing descriptions in archival objects, while taking advantage of the CIDOC-CRM entity where they can aggregate. In our extension of CIDOC-CRM, named ArchOnto, new properties have prefix "ARP" and new entities prefix "ARE".

Table 1. ISAD(G) to CIDOC-CRM and extensions

ISAD(G)	CIDOC-CRM	CIDOC-CRM extension
Administrative/biographical history	P3 has note	ARP1 has administrative history
Archival history	P3 has note	ARP2 has archival history
Scope and content	P3 has note	ARP3 has scope
Conditions governing access	P3 has note	ARP4 has access conditions

In addition to these Data Properties, new Object Properties were also introduced. They capture the hierarchy of levels of description in ISAD(G). Table 2 shows the two properties used to represent the relations between levels of description. The properties are the inverse of each other, and their instances capture the relations between levels. For example, a Fonds is the top of the hierarchy and can relate to a Sub-fonds via ARP9 lower level, but not to a upper level. Similarly, a document can have a Series as an upper level, or a Process.

Table 2. ISAD(G) to CIDOC-CRM - new object properties

Description	CIDOC-CRM extension
Representation of upper level	ARP8 upper level
Representation of lower level	ARP9 lower level

A new class was also introduced to represent the level of description, ARE1 Level of Description. This class, a subclass of E55 Type, is instantiated with the description levels used in the archives, that actually provide a type to each of the archival objects.

The new properties and entities presented here are a sample of what has been proposed for the ArchOnto model[6].

[6] OWL version available on GitHub: https://github.com/feup-infolab/archontology.

4 A Prototype CIDOC-CRM Knowledge Graph

We carried out an in-depth analysis and prototyping in order to design an architecture for the information system that will implement the CIDOC-CRM compliant knowledge graph[7]. For this purpose, we have decided to adopt a graph database because they have a simpler, more interoperable model that naturally integrates in the LOD cloud, one of the chief requirements of the whole endeavour. If we chose a relational database instead, and kept the plain semantic web model, there would be a systematic need for complex queries requiring costly `JOIN` operations, as the database would most likely degenerate into a very large table for the edges of the graph.

Additionally, the fact that CIDOC-CRM was designed according to object-oriented programming (OOP) [6] principles makes the model suitable to the application of an Object-Graph Mapping (OGM). This technology resembles Object-Relational Mapping for relational databases, and makes basic CRUD (Create-Read-Update-Delete) operations transparent to the programmer.

The ANTT specified the requirement that all components of the system should be free from any licensing costs and adopt open-source licenses. With this in mind, our database of choice was Neo4j, because of its open-source license (GPLv3), wide adoption, support for ACID transactions and most importantly a mature OGM library[8] that runs on the Java Virtual Machine (JVM). This makes the solution cross-compatible with any operating system, enabling the ANTT to adopt Linux-based servers for its archival information system.

4.1 Multiple Inheritance: the Diamond Problem

One of the first issues encountered when attempting to interface with the graph through an OGM was the fact that the CIDOC-CRM defines a complex set of classes and subclasses, where there are many cases of multiple inheritance. In OOP, this is a classic problem known as the "Diamond Problem"—when a class inherits from two or more other classes.

To maintain a clear code organization as close to the CIDOC-CRM concepts as possible, we needed a programming language capable of supporting multiple inheritance in an elegant way, while also supporting the Neo4j OGM, which is designed to run on Java. Our choice was Groovy, a modern language mostly compatible with Java syntax, which approaches the diamond problem through the use of "traits": sets of methods and fields akin to Java interfaces, which can be added to one or more classes to represent multiple inheritance. The Groovy compiler resolves method naming conflicts by using the method last declared when multiple other traits are extended. However, should one wish to use the same named method from a trait that was not the last declared one, this is still possible through additional configuration options.

[7] Prototype available on GitHub: https://github.com/feup-infolab/archgraph.

[8] Documentation: https://neo4j.com/docs/ogm-manual.

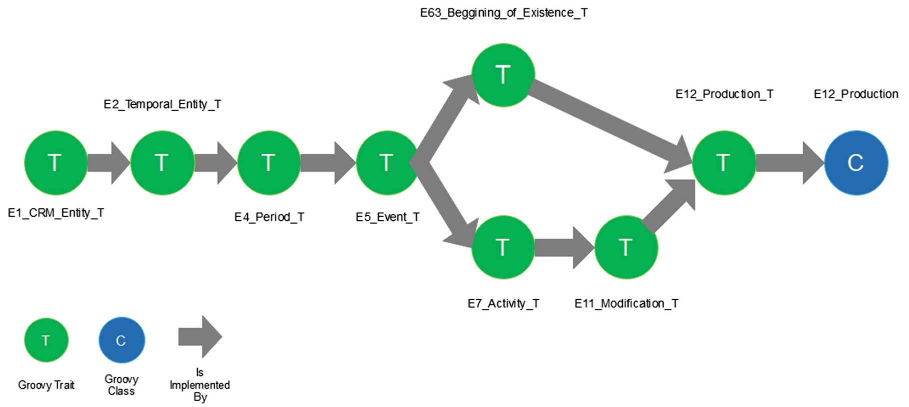

Fig. 1. Implementation of the traits for class E12 Production

The diagram in Fig. 1 is a graphical representation of the implementation of traits in class `E12 Production`. Several Groovy traits are created for each CIDOC-CRM class, and then combined at will to represent the multiple inheritance cases: in the example, `E12 Production` is a subclass of both `E11 Modification` and `E63 Beginning Of Existence`, both modelled as traits. Every trait contains its own `rdf:type` and its own methods. In this case the `E12 Production` "leaf" class is the only one that is to be instantiated. When an instance is created, it automatically gains all the methods in its trait hierarchy. When it is persisted to the Neo4j database, all `rdf:type` will be recorded, from `E12 Production` up to `E1 CRM Entity`.

4.2 Validation Trade-Offs

In theory, a graph data model should comply with an ontology and therefore be validated for consistency on every update, ensuring consistency with the model. In an operational system, however, such operations are not feasible due to their high computational cost. As a result, while developing the database and utilizing the OGM, one of the major concerns was to make sure that the model required as little external validation for consistency as possible. Using CIDOC-CRM and the OGM from Neo4j, we try to carry out most validation by enforcing, for each entity, the type induced by the corresponding class and ensuring that the relationships have the proper domains and ranges as defined for their properties. This includes making sure that the domains and ranges are subclasses of the classes specified as such on the CIDOC-CRM.

To further alleviate the validation effort, we decided to create extensions to the CIDOC-CRM, specifically for the software implementation of data properties and of ternary associations, which in CIDOC-CRM are denoted as properties of properties.

The new data properties were created with the prefix "ARP" and ternary associations were implemented by creating node entities that represent the property and have the prefix “PC”, then creating properties `ARP01 has domain` and `ARP02 has range`. This allows the implementation of ternary relationships without any further validation besides compile-time verification.

Figure 2 is an example of a sub-graph showing an archival record for the Portuguese law that made the adoption of the Gregorian calendar official. This graphical representation is generated directly by Neo4j as an output of the existing version of our software and data model.

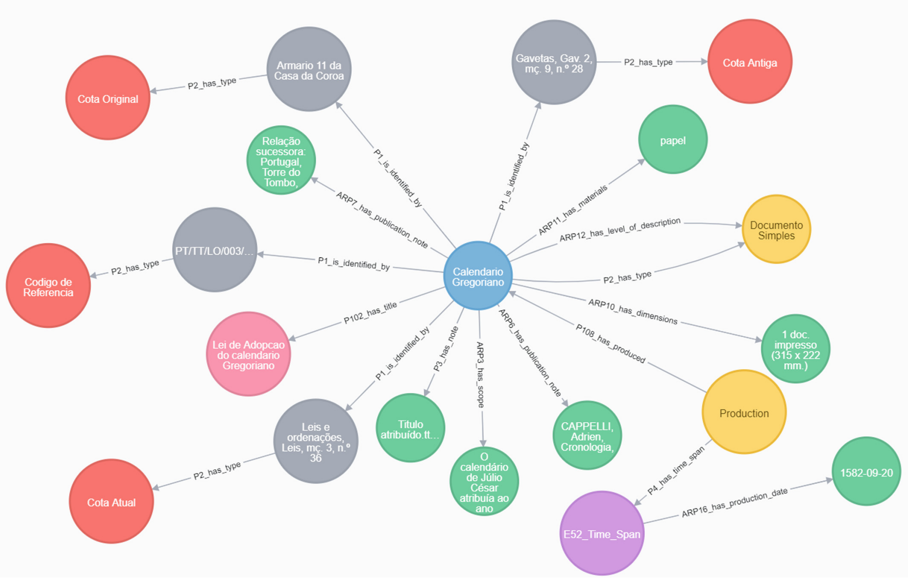

Fig. 2. The Portuguese law for the adoption of the Gregorian Calendar, as a graph (Color figure online)

This graph represents the neighbourhood of the “Calendário Gregoriano” resource (Gregorian Calendar). It represents the law that passed the adoption of the Gregorian calendar in Portugal. The Gregorian Calendar record is at the center, in blue. It is linked to the physical material of the record (“papel” or paper), as well as publication notes, the title scopes, general notes, dimensions and level of description. Several objects of assertions for `P2 has type` (in red) are the original, the previous and the current call number for the record. Finally, the record holds a reference code and is linked to its production date.

5 Conclusions

In this paper, we present several challenges behind the implementation of a knowledge graph at the Portuguese National Archives. Balancing the expressiveness of the graph model and the demand for high performance in an operational system, while ensuring elegant code organization, have led us to complement the existing CIDOC-CRM with new classes and properties and to adopt the Neo4j graph database, the Groovy programming language and the Grails framework.

The initial approach to the archival model and corresponding knowledge graph allowed us to identify issues that will be addresses in the next steps and have to do with graph traversal, querying and searching the database and user interfaces.

Cypher, the Neo4j query language, is currently being used mostly for testing. In the future, the OGM will handle CRUD operations, traversals and searching, while a few custom Cypher queries will handle complex cases.

Query and traversal modules are not implemented and tested yet; should the OGM fail to satisfy all the requirements, it may be complemented with a graph traversal Domain-Specific Language such as Apache Gremlin, more suited to run graph traversal and thus act as a connector for the machine learning modules required by the migration and information extraction tasks in the project.

The user interfaces have to satisfy the diverse requirements of applications and end users. For applications, the graph must provide CRUD operations, as well as the display of sub-graphs. Users can be professionals (archivists creating new records), or other users and all need to access the information on the records in an intuitive manner, while traversing the graph based on archival contents and relationships. For the business logic we will use Grails as the web framework, which in turn uses Groovy, already in use for interacting with the database. Additionally, Grails has plugins that can be used with Neo4J to allow a better integration with the rest of the application, and can also share the same server.

This work is part of a larger project, EPISA, where a systematic migration from ISAD to the new model is expected, together with domain-specific data mining on the extensive textual fields in the records. The project also aims to satisfy requirements from different stakeholders: archivists who need a convenient environment for description; scholars who use the archives as their main data source; and laypeople who explore the archives for various purposes.

References

1. de Almeida, M.J., Runa, L.: ICON project: content integration in Portuguese national archives using CIDOC-CRM. In: 2018 CIDOC Annual Conference (2018)
2. Bountouri, L., Gergatsoulis, M.: Mapping encoded archival description to CIDOC CRM. In: First Workshop on Digital Information Management, Corfu, Greece (2010)
3. Direcção Geral de Arquivos, Grupo de Trabalho de Normalização da Descrição em Arquivo: Orientações para a Descrição Arquivística, 2nd edn. Direção Geral de Arquivos, Lisboa (2007)
4. ICOM/CIDOC CRM Special Interest Group: Definition of the CIDOC Conceptual Reference Model. ICOM, 6.2.2 edn. (2017)
5. International Council on Archives: ISAD(G) Second Edition. ICA (2000)
6. Oldman, D.: The CIDOC Conceptual Reference Model (CIDOC-CRM): PRIMER. CRM Labs (2014)