

Information Systems for IST Museums

João Afonso Pereira, 78949, MEEC
Instituto Superior Técnico, Universidade de Lisboa
Lisboa, Portugal
e-mail: joao.pina.pereira@tecnico.ulisboa.pt

Abstract—Currently, most museums are supported by a digital base, even preserving a physical documentation support. An approach that is not effectively implemented in the Instituto Superior Técnico museums, either in Web applications and portals, or in a consistent and reliable database. Thus, the objective is to study how to evolve using current technology.

It was thus developed - through a museum framework: Omeka S - a module connected to software called ISTCIDOC. This module allows implementing a series of functionalities, among which: to standardize, according to CIDOC norms, an ontology proper for the DECivil museum; and a series of graphical user interfaces for the museum to create/edit/delete an item with its metadata. In addition, a new entity was created in Omeka: Location, following the implementation in an online library of the spaces located in the Technician. This module was based on other modules available for Omeka S, including its own internal module.

Thus, as objectives of this work, we have the future integration of the DECivil museum in the Portuguese Museum Network. Something that will help in better dissemination and recognition of the museums of the Instituto Superior Técnico and bring more awareness to the population of the existence of heritage of this faculty.

Index Terms—database, Omeka S, module, metadata, Portuguese Museum Network, museum DECivil.

I. INTRODUCTION

With the arrival and expansion of the Internet, we began to question the benefits of new technologies when in contact with museums. Since 1997, an annual conference called “Museums and the Web” [1] dedicated to the subject takes place in North America, resulting in many publications and collaborations. We know that museums, even those of recent times, have antique and collectible articles, in which a conventional method of storing their articles and all their archived history persists.

Therefore, the self-existence of a virtual museum should provide means to visit a museum that cannot be visited in the real world, either because it is difficult or because this virtual space does not exist, as well as complementing the real museum giving the ability to users to do something that is not possible, how to manipulate and manipulate objects freely and/or measure them.

Regarding the dissemination of a specific article or the museum itself, it is fundamental to have a database in which it is possible to store all items with their respective information and history, so that anyone can want to meet and visit any museum more easily and accessible to everyone by doing research on the Internet.

These lessons learned underscore the importance of adopting a visitor-centric approach in developing digital museum resources and the need for museum researchers and practitioners to better understand how new information technologies have changed the way museum visitors approach museums and museums. their resources [2].

There are several ways to store inventory records, namely in the digital domain, the most common and traditional being the use of spreadsheets such as Excel. Other software tools include Omeka (Classic or S), Collective Access, CollectionSpace, OpenExhibits, and Pachyderm. These are the top five free and open source tools. Finally, you can use a database as simple as MySQL (note that *frameworks* make use of languages to manipulate databases: SQL, DQL, etc.).

A. Problems with existing Solutions

Nowadays in the digital age, it is important for any infrastructure, whatever it may be, to have a built-in digital base, such as website and database. The fact that museums have a very strong and important traditional historical part means that the entries of new technologies for the system are resisted by them, but as in everything else in life, there is always space and there will have to be change and adaptation to this introduction.

The present analysis aims to reflect on the development and adaptation of cultural communication to the context of digital media and to show its use in strategies developed in a museum environment [3] that “can help solve the problems of accessibility - not only geographic and economic, but also intellectual and behavioural of people. Its ability to provide unprecedented access to collections and knowledge, to contextualize, customize and adapt content and enable interactivity and feedback, makes them the ideal tools to create a better relationship with communities of interest and create new ones. experiences that attract new audiences.” [4].

Another problem that occurs is the difficulty of maintaining and making safe all the information and factsheet of the objects of the museums. Even adopting a more digital approach, as it is currently done in the DECivil museum - namely in Word/Excel format - there is a great lack of versatility and it is a very unintuitive solution for any user, as will be described in topic that follows.

To complement what has been said, there is a current lack of other technologies, such as the use of apps, in which this work does not focus, but nevertheless it is worth noting all the

importance it has for the development of the transformation to the digital domain by part of museums in general.

There is an extreme difficulty in performing all the different aspects that are essential for storing inventory tokens from a museum. First, the part of backend: save and/or manipulate items in the database (with MySQL and PHPMyAdmin, for example). Second, the part of the frontend: dealing with web pages, with all the necessary CSS, JS and HTML. Third, the connection between the two that is made through PHP.

Therefore, there are *framework* solutions mentioned in the *framework*, which include all these topics and allows for an easier and more intuitive integration of everyone.

B. Proposed Solutions

The solution involves the use of a generic software development *framework* that can be applied to most museums and be of assistance in planning the system and information needs. This is because a *framework* is based on functional descriptions of museum activities and the information and precise relationships for all system support [5].

Given the brief introduction to this subject, the Omeka S *framework* was then used for this work. It allows solving the problem described above and doing everything with a very visual and easy to work, no programming skills. The scope of the objective of the thesis will be to integrate the catalog of inventory of the Museum of Civil of the Technician with parameters based on an International norm and standard of inventorying items in museums: CIDOC. For this, it was necessary to create a root module for Omeka S, which allows the creation of an item according to these standards - with attributes other than those installed (DC, Darwin Core, BIBO, etc.), discharging a new adapted ontology called ISTCIDOC (RDF/OWL file). In addition to new attributes and classes (from this new vocabulary), a new entity is created that is the Locations, referring to Instituto Superior Técnico installations. These are also stored in the same Omeka database and you create or select a location by integrating (in frontend, in the PHP part of some module files) the Fenix API (namely the GET/spaces/{id}).

In addition, Protégé software was used to create the modified ISTCIDOC ontology with the new classes and attributes, and to generate an RDF/OWL file. The EasyRDF plugin, to read this file that has the vocabulary and write to the Omeka database, in PHP within the module when it is installed. All this process of making the module was done in PHP, HTML and even a little JavaScript and CSS, namely in the part of the integration of the API of Fenix.

In this way, it is possible to create an efficient and organized digital platform in which the data of the museums are stored in the system and exposed online, which allows, as previously mentioned, a quick and effective exhibition of the museums, in this case the Technician.

II. THEORETICAL AND TECHNICAL BACKGROUND

This chapter contains essential topics related to this work. Firstly, the CIDOC standards, with the relevant classes and

attributes, and the brief description of the CIDOC CRM will be explained. Then the Omeka *framework* is explained: related modules, installed vocabularies, and analysed other similar *frameworks*, in which this work could have been done as well.

A. CIDOC

CIDOC (International Committee for Documentation), an International Committee for Documentation from one of the 30 international committees of the International Council of Museums (ICOM), is an international, non-profit, non-governmental organization, dedicated to the documentation of museum collections, being also used for the development of records and allusive to the good practice of keeping them.

According to the International Guidelines for Museum [6], a set of objectives proposed by CIDOC should be enumerated, such as:

- Ensure responsibility for objects: these can be used to define objects that are owned by a museum, identify objects and record their location;
- Helping the safety of objects: these can be used to maintain information about the status of objects and provide descriptions and evidence of property in case of theft.

Also, should be defined CIDOC CRM. This document [7] is the formal definition of the CIDOC Conceptual Reference Model (“CRM”), a formal ontology model designed to facilitate the integration, mediation and exchange of heterogeneous cultural heritage in formation. CRM is the accumulation of more than a decade of ICOM’s CIDOC standards development work [7].

The “CIDOC Object Oriented Conceptual Reference Model” (CRM) is a domain ontology but includes its own version of a higher ontology. The main classes contain (few cases) [8]:

- Space time: includes title/identifier, location, era/period, time interval, and relationship to persistent items.
- Events: Includes title/identifier, beginning/end of existence, participants (people, individually or in groups), creation/modification of things (physical or conceptual) and relationship with persistent items.

Examples of definitions (few cases) [9]:

- Persistent Item: a physical or conceptual item that has a persistent identity recognized within the duration of its existence by its identification and not by its continuity or observation.
- Temporary Entity: includes events, eras/periods, and condition states that occur for a limited period and are separated from the Persistent Item. A Temporal Entity is comparable to a lasting one.

B. Museums

The most important in this case will be to catalogue each item (object record) of a designated Technician museum, of the four currently existing. The museums in question are: the DECivil museum (from the Department of Civil Engineering, Architecture and Geosciences [10]), the Faraday Museum (in the electricity pavilion - museum opened in 2016) and two Geosciences museums (located on floors 2 and 3 of the Minas

Pavilion). The focus for this work will be the DECivil museum, since one of the coordinators - teacher Ana Tomé - is involved in this project as well.

The collections of items with the DECivil museum tokens can be any type of date - they can be changed, added and deleted at any time - but following the typology of class types and properties imposed by the new ISTCIDOC ontology.

Therefore, for the final presentation of the museum, in conjunction with Omeka S, it will follow the following topology, from three different types of “building blocks” [11]:

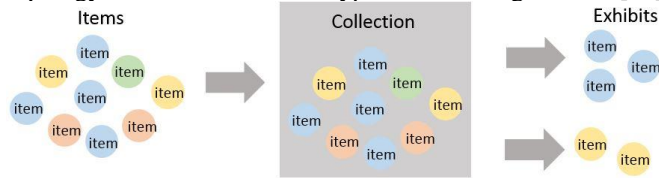


Figure II.1: Topology of the final view of a museum in Omeka.

C. DECivil Museum in RPM

The Portuguese Museum Network (RPM) and an organized system of museums, based on voluntary membership, progressively configured and aimed at decentralization, mediation, qualification and cooperation among museums [12].

Currently, the Portuguese Museum Network is made up of 151 museums. The richness of its universe lies in the diversity of tutelars, collections, spaces and facilities, educational and cultural activities, models of relationship with communities and management systems [12].

During this project, the following goals (to be explained in more detail in the following sections) and the definition and implementation of the Information Systems needed by museums should be aimed at:

- Integrate the Civil Museum into the Portuguese Museum Network
- Civil works as physical museum → bet on Virtual Exhibitions
- Three different models given for each museum → different colours of Omeka
- Manage tokens and catalogue of parts and objects
- Presentation of the information of the present and catalogued collections
- Virtual representation (on the web) of exhibitions that take place
- Providing information about the various parts.

D. Omeka Classic and S

Omeka is a free and open source content management system for online digital collections [13]. This software allows the creation and modification of items of objects of a museum; the creation of pages and sites of a museum; use of several ontologies and vocabularies. There are currently two versions of Omeka, the Classic version, the S version.

Omeka's default metadata model is the Dublin Core. A site from Omeka (Web interface for the visitor) includes [14]:

- **Items:** links to a navigable list of items, selectable by item type and tags.

- **Collections:** Groups of items, the public can dig through the collection to find items.
- **Exhibits:** exhibits contain interpretive texts and depend on items/sources/objects as building blocks.
- **Keyword search box:** option to link the advanced search page.

Omeka Classic is a web publishing platform for sharing digital collections and creating rich online exhibitions on average [15].

Omeka S also incorporates a developer's documentation for its website.

E. Omeka Modules

This section refers to type modules used for a better understanding and application of reverse engineering techniques to complement the documentation provided by Omeka for developers. This is because, with only existing documentation now, it is not possible to write a new root module.

This includes three based modules which are named: Custom Ontology, Collecting and Custom Vocab. Those modules serve a certain purpose for each part of the new module development.

F. Internal Module of Omeka S

The Omeka S system itself is managed by the internal module itself. Much of this work was possible due to the reverse engineering of this module, namely in the essence and structure of the module creation. This, like all other modules, follows a *framework* typology of Zend Framework 2, a *framework* that will be described in more detail in section III. This has Omeka DQL associated to be briefly explained.

Essentially, DQL provides powerful query features about the object model. This is a scenario in which all objects are scattered in some storage (such as an object database). When writing DQL queries, one should consider consulting this store to choose a subset of its objects [16].

G. Vocabularies of Omeka S

Here you will find some standard vocabularies (i.e. pre-installed) of Omeka S. These vocabularies are a collection of RDF metadata properties and properties published to describe a resource. These exist and are created externally for Omeka and can be imported (with some limitations) to the Omeka S for use throughout the facility. The most commonly used vocabulary is the Dublin Core Terms (“dcterms:”) [17].

Global and site administrators can manage installation vocabularies in the “Vocabularies” tab in the navigation to the left of the Admin panel. Within Omeka, it is possible to do the following with vocabularies [17]:

- **Add vocabularies:** with a prefix, URI, tag, commentary, and the file itself.
- **Edit vocabulary.**
- **Import an existing vocabulary directly** - usually in the RDF or OWL file format.
- **Delete one or several vocabularies.**

Now, it's going to be described several vocabularies, such as Dublin Core, Darwin Core and Bibliographic Ontology.

Dublin Core is a metadata scheme that aims to describe digital objects such as videos, sounds, images, texts and websites on the Internet. Dublin Core applications use XML and the Resource Description Framework (RDF).

The Darwin Core (often abbreviated to DwC) is an extension of the Dublin Core for biodiversity computing. It is intended to provide a stable standard reference for sharing information on biological diversity [18].

The specification of Bibliographic Ontology (BIBO) provides the main concepts and properties to describe citations and bibliographical references (i.e. citations, books, articles, etc.) in the Semantic Web. This is written in RDF and can be used as a citation ontology, as a document classification ontology, or simply to describe any kind of document in RDF.

This vocabulary contains properties such as: Creator, Description, Identifier, Title and Editorial Note. In addition, it includes classes such as: Event, Agent, Academic Article, Book, among others.

H. Comparison with other frameworks

The top five free, open-source tools for creating and managing virtual views are Omeka, Collective Access, CollectionSpace, OpenExhibits, and Pachyderm [19]. These frameworks have their advantages and disadvantages, and there are cases where one is more focused on one thing, or one framework is more generic and less detailed.

Note that in the case of Omeka, we're assuming this was using the DC (Dublin Core) ontology. In many cases, Omeka's DC does not get many of the attributes of CIDOC, in a total of only 22 class types and 55 properties. Hence the need to import a RDF ontology of CIDOC, or within the scope of this dissertation, a custom IST-CIDOC ontology, which will be described later.

III. DESIGN

This chapter presents an analysis of the requirements/accesses, objectives and general design (in relation to the ISTCIDOC ontology) intended for this dissertation. Above all, it will address fundamental aspects such as the description of the new ontology.

A. Access Control

An entity that must be considered are the users, i.e. museum visitors. In order to test this work, because it is currently in the development phase and not yet in production, the portal was done on a local server (designated by "Localhost", omission address 127.0.0.1) through a program called Apache (HTTP server, responsible for more than a dozen projects involving technologies of webcasting, data processing and execution of distributed applications [20]). Although, with the help of Omeka S, it is possible to pass the local effects to an external HTTP server, with a Technician domain.

Users should then have access to the collections, in case they are public or if the latter's permission dictates that they can visit as well. In addition to the various collections, they can

have access to possible virtual exhibitions, in which Omeka S makes available in an accessible and incorporated way in the framework itself.

Omeka S defines a set of user types, in which they have different permissions. There are six distinct roles, which are [21]:

- **Global Administrator:** complete installation privileges.
- **Site Administrator:** Robust Site and Content Privileges.
- **Editor ("Content Expert"):** Total privileges for content creation.
- **Reviewer:** Privileges of robust content but can only exclude the content itself.
- **Author:** create your own content.
- **Researcher:** research and read only privileges.

B. CIDOC for IST

In this work, and starting from an initial phase, we considered the set of classes of the new ontology from a class defined by CIDOC CRM, called E70 Thing. Here are two types of objects: E18 Physical Thing (physical object) and E28 Conceptual Object (conceptual object). The first is defined as something material and the second case as immaterial.

All this is possible to retain through the diagram shown in Figure III.1 (extracted from [22]).

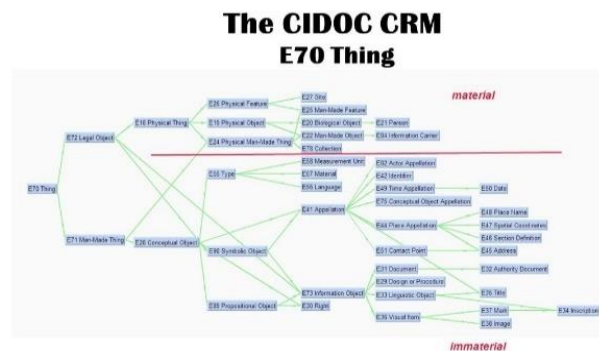


Figure III.1: CRM Diagram of the entity E70 Thing.

The following list corresponds to the classes for the ISTCIDOC ontology: Accessory; Sample; Demonstration; Apparatus; Computer; Drawing; Sculpture; Architectural; Sculpture; Sculpture of Volte; Funeral Sculpture; Tool; Model; Furniture; Model; Pedagogical; Painting; Watercolour; Oil; Pastry; Mural; Hardening; and finally, Stained glass.

At last, the ISTCIDOC properties are presented next: Identifier; Title/Name; Description; and Location (Instituto Superior Técnico's Location).

So, the ISTCIDOC ontology has 21 classes and 4 properties.

IV. ARCHITECTURE AND IMPLEMENTATION

This chapter presents all the provisioned architecture of the ISTCIDOC module, as well as all support systems and technologies used. Topics such as module structure and HTTP requests that are involved in it will be covered. Then follows

the implementation part, with the description of the two entities created, the interfaces that make up the module, and also a brief analysis of how the Omeka database is processed.

A. Protégé

Protégé is a free and open source ontology editor for building intelligent systems [23]. Protégé allows the reuse of *frameworks* for the construction of domain models through its support for declarative domain ontologies.

Visual notation for OWL (VOWL) then forms part of a Protégé plugin, where it is possible to observe, through Figure IV.1, a part of the CIDOC CRM ontology (version 6.2.1). The circles represent the classes and the links by arrows are the properties.

Each class and properties can have multiple fields for addition, as in the case of subclass annotations or sub properties, and domains and scopes, in the case of object properties. After creating the classes and properties of a new ontology, you can save and export the ontology to a format typically of RDF/XML or OWL/XML.

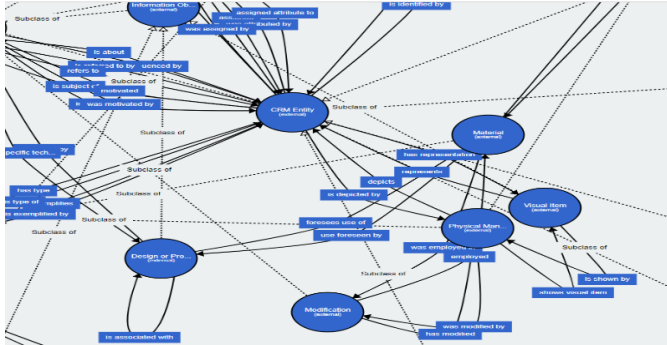


Figure IV.1: VOWL Diagram of part of the CIDOC ontology.

B. Zend Framework

The Zend Framework will be the *framework* used for the module structure of this work. All modules that are part of Omeka, including the internal module itself, must follow the architecture and structure implemented by ZF.

Zend Framework (ZF) is an open-source, object-oriented Web application *framework*, implemented in PHP 7 and licensed under the New BSD License [24].

In the scope of this work, the successive test of the new ISTCIDOC module was not, however, accessed by this *framework*, but directly by Omeka. Only at an earlier stage was an example module tested to help familiarize me with the module structure and model. Note that it was still relevant to have this *framework* in mind, in the sense that Omeka itself is built from there.

C. EasyRDF

EasyRDF is a PHP library designed to facilitate the consumption and production of RDF [25].

After parsing (initial processing), EasyRdf builds a graph of PHP objects that can be moved so that the data is placed on the page. Dump methods are available to inspect what data is available during development. Data is typically loaded into an

EasyRdf\Graph object from source RDF documents, loaded from the Web via HTTP. The EasyRdf\GraphStore class simplifies the loading and saving of data in a SPARQL 1.1 graphical storage.

For the ISTCIDOC module, this plugin was used to extract the classes and properties from the OWL file created, called "ist-cidoc.rdf". The process of creating and developing this file with the new ontology will be described later.

D. IST FenixEdu API

This section of the Fenix APIs is part of FenixEdu™. FenixEdu™ is a software project focused on developing open source software for schools. The main development team currently works at Instituto Superior Técnico [26]. It covers a range of solutions, from academic and administrative processes to general purpose software libraries.

The page with the FenixEdu API list has a series of endpoints, which can be used for several utilities and compatible with several languages, among which PHP, in which this project is written about.

Therefore, in this work only two endpoints will be used: the main one being the GET/spaces/{id} protocol, and the GET/spaces/{id}/blueprint protocol. The first one strips the information of a specific room, considering the starting point the Alameda Campus (ID: 2448131360897). This protocol will be better described below. The second case is only used to extract the URL of the image from its room or space.

The GET/spaces/{id} API was then used for the 'add-location.phtml' file, where a parameter 'ID' (id = ...) of the GET request of the URL is made. If the ID is invalid or non-existent, then by default, the page goes to the starting point, which will be the Alameda Campus. Without wishing to go into too much detail here (as will be explained better in the interfaces of section 5), the GET/spaces/{id} request is used by calling the SDK function: getSpace(\$spaceID) – in order to create a JSON variable with all the information about that space.

The protocol GET/spaces/{id}/blueprint is also used to obtain the image of the space itself, to which the user is currently (this {id} is checked through the generated JSON variable subfield).

E. ISTCIDOC Module Structure

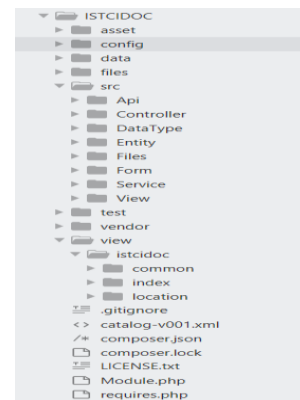


Figure IV.2: Directory structure and files of ISTCIDOC module.

Within the ‘src’ folder, all the PHP driver files and other types are present, in order to manage the entire backend part of the module; as opposed to the ‘view’ folder, in which are the PHTML files (PHP with embedded HTML) with the frontend area, that is, where the end user (with Omeka permissions) accesses the page portal. The config files are contained in the ‘config’ folder.

As mentioned earlier, the file ‘Module.php’ manages the installation/uninstallation of the module and the resulting actions.

F. Module Architecture

The diagram in explains the architecture of the entire module, in which the linking of locations to items is noted. This is since, since one of the parameters (properties) of ISTCIDOC is the location where the item is located, it is integrated into the metadata of the ISTCIDOC item itself.

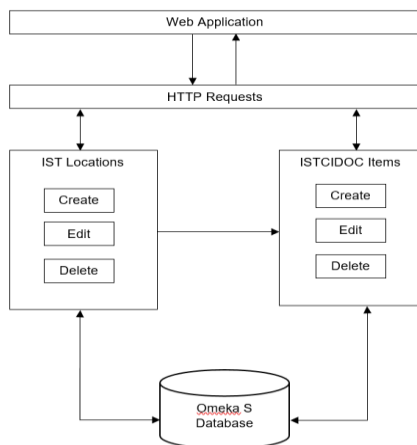


Figure IV.3: Diagram with the module architecture.

The idea is that the module has a set of interfaces in frontend, and that after processing HTTP requests (either by GET or POST), they access the Omeka S database, and one or more responses are sent back to the user, whether for creating, editing, deleting or viewing items and locations.

G. HTTP Requests

The Table IV.1 depicts all the HTTP requests used:

Action	Request URL	Request Method	HTTP Status Code
Add ISTCIDOC Item	../admin/ist-cidoc	POST	200 OK
Edit ISTCIDOC Item	../admin/item/{id}/edit	POST	302 Found
See ISTCIDOC Items	../admin/ist-cidoc/browse	GET	200 OK
Delete ISTCIDOC Item	../admin/item/{id}/delete	POST	302 Found
Add Location	../admin/ist-cidoc/add-location?id={id}	POST	302 Found
Edit Location	../admin/ist-cidoc/location/{id}/edit	POST	302 Found
See Locations	../admin/ist-cidoc/browse-location	GET	200 OK
Delete Location	../admin/ist-cidoc/location/{id}/delete	POST	302 Found
Search Locations	../admin/ist-cidoc/browse-location?{query_parameters}	GET	200 OK

Table IV.1: Description of HTTP requests and responses, for each functionality.

H. CIDOC Implementation for the objects of IST Museums

Connecting the CIDOC CRM ontology, the classes of this object would then have the following format (identified by the number 1 to 16 to match the properties to the front - the attributes are all in English, in the original format):

1: Part Title -> Man-Made Object (E22): Goniometer; 2: Part description -> Man-Made Object (E22): Goniometer w/2 Mirrored Lunettes - Vertical; 3: Inventory -> Identifier (E42): A - 17M; 4: Inventory Date -> Birth (E67): 1995; 5: Photography -> Image (E38): Fig. 1; 6: Dimensions -> Dimension (E54): Height; 7: Measurement of Dimension -> Measurement Unit (E58): cm [centimetre]; 8: Dimensions -> Dimension (E54): Diameter; 9: Dimensions -> Dimension (E54): Width; 10: Dimensions -> Dimension (E54): Weight; 11: Dimensions -> Dimension (E54): Depth; 12: Material -> Material (E57): Platinum, Metal, Glass; 13: Location -> Place (E53): Minerology - Department of Mine; 14: Supplier -> Person (E21): R. Fuess Steglitz - Berlin; 15: Quantity -> Number (E60): 1; 16: Remarks -> String (E62): Cabinet Meeting Room (Missing eyepieces, see 22M Fig. 25).

The properties are then as follows (referring to Man-Made Object (E22): Goniometer, and identified from ‘A’ to ‘G’ - also in English of origin):

A: has number of parts (P57) -2; B: is identified by (P1) -3; C: brought into existence (P92) -4; D: Visual shows item (P65) -5; E: has unit (P91) -6 -> 7; F: has unit (P91) -8 -> 7; G: consists of (P45) -12; H: has current location (P55) -13; I: supplied by -14; J: has number of parts (P57) -15; K: has note (P3) -16.

Form fields include: Folha de Inventário, Nº de Inventário, Data do Inventário, Título do Objeto, Descrição do Objeto, Fotografia, Dimensões (Altura, Largura, Profundidade), Material, Coleção, Localização, Proveniência, Marca, Quantidade, Estado de Conservação, Data da Fotografia, Observações, Histórico Data de Aquisição, Motivos de Aquisição, Proveniência, Exponção em que foi exibido, Bibliografia em que é citado, Avaliação, Estado Avaliado, Legenda A - referência ao inventário antigo; B - referência ao inventário em curso.

Figure IV.4: Object tab of a Goniometer of Instituto Superior Técnico.

I. CIDOC CRM for Omeka

CRM is a formal ontology that can be expressed in terms of logic or a language of adequate knowledge representation. Their concepts can be instantiated as sets of statements that provide a model of reality. Thus, the data expressed in any form compatible with CRM can be automatically transformed into any other form compatible with CRM without loss of meaning.

CRM classes and properties are identified by their initial codes, such as “E55” (“Type”) or “P12” (“occurred in the presence of, or was present in”).

An RDF file (.rdf extension) will then include classes, which can have associated subclasses, super classes, a scope note, a tag, and so on; and properties: in which may be associated sub properties, super properties, a domain (which will be the class associated with that property) and scope - which corresponds to the target class in which it contains all the potential values of a property, and inheritance.

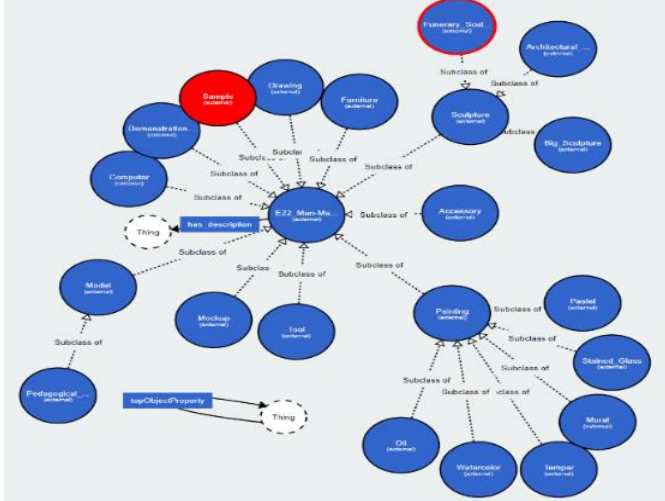


Figure IV.5: OWL diagram of the ISTCIDOC ontology.

Through the software mentioned in section A, classes and properties were added for the new ontology. First, an IRI of Ontology (“Ontology IRI”) had to be defined. Each ontology can have an IRI ontology, which is used to identify an ontology [27]. There is the possibility of having an IRI version as well, but in this case, it has not been defined. The IRI used will then be this URL: <http://www.semanticweb.org/miguelafonso/ontologies/2019/0/ist-crm>.

J. Entity ‘istcidoc_item’

This new entity is part of a Resource and has the following attributes:

- A class of this object.
- Unique identifier.
- Title/Name.
- Description.
- Location (previously added by the user and belonging to Instituto Superior Técnico).

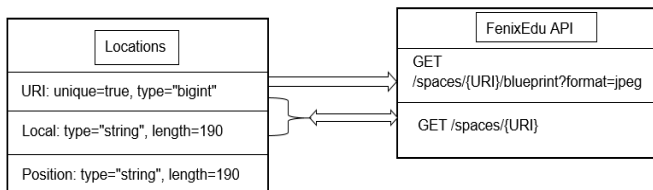


Figure IV.6: Diagram with the attributes of Location and API integration with FenixEdu API.

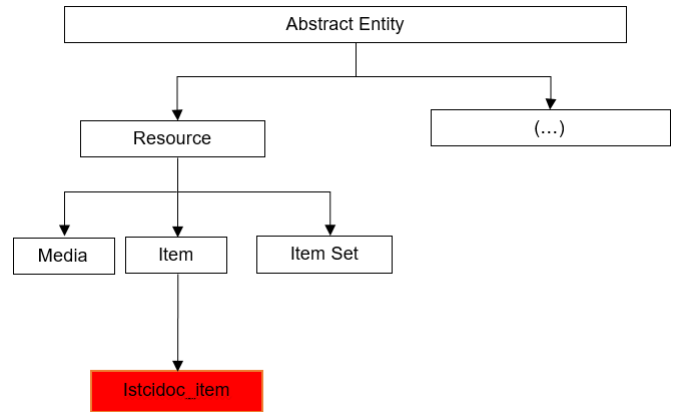


Figure IV.7: Diagram with the entities structure until ‘istcidoc_item’.

K. Entity ‘Location’

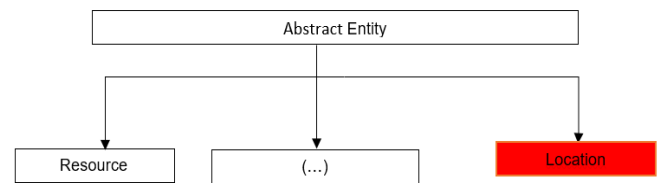


Figure IV.8: Diagram with the entities structure until ‘location’.

As can be seen from the diagram in Figure IV.8, this entity is an extension of the Abstract Entity, that is, it is associated with the top of the hierarchy side by side with the Resource.

The Figure IV.6 shows all attributes/properties of the Location entity (or Location for Omeka S) with the data types that each field supports. It is important to note the incorporation of the FenixEdu API with the URI and Local parameters, and for the GET request of the space, it can be used in both directions. However, for the GET request of the image, it will only be provided by the parameter and URI, not in the reverse direction.

L. Access Control List

In relation to the ISTCIDOC module, this was defined for an access control list (called ‘ACL’) for privilege management [28].

The ACL permissions associated with each module page, are called through a function “allow (...)”. If the first argument of the function is NULL, then the role is assigned to any type of user. This means that any user has access to see the list of items and the details of each, as can be seen from the first call to the function.

The same procedure is analogous for the remaining function calls.

M. ISTCIDOC Interfaces

Here all the interfaces implemented in the module will be announced. The topics presented are as follows (as shown in Omeka S):

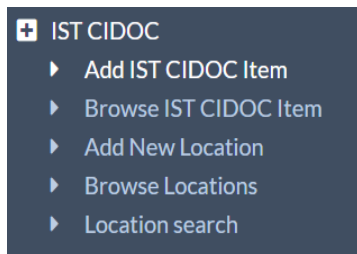


Figure IV.9: ISTCIDOC module interfaces.

Next, is briefly shown the images for each interface:

1) Add IST CIDOC Item

 A web form titled 'Add IST Item' with a 'Submit' button. It contains several input fields: 'Identifier' (with a dropdown 'from Identifier'), 'Title/Name' (with a dropdown 'from name/title'), 'Description' (with a dropdown 'description of the item'), and 'Location' (with a dropdown 'location of the item'). There are also 'Text' buttons next to the Identifier, Title/Name, and Description fields. A 'Select Location' sidebar on the right shows a list of locations, with 'Gabinete Alameda' selected.

Figure IV.10: Interface to add new item.

2) Browse IST CIDOC Item

 A table titled 'ISTCIDOC Items' with a 'Add new item' button. The table has columns: 'Item Title', 'Classe', 'Dono', and 'Created'. It shows a list of items with their titles, classes, owners, and creation dates.

Item Title	Classe	Dono	Created
22222	Accessory	MuseusIST	Feb 11, 2019
[Sem título]		MuseusIST	Jan 31, 2019
[Sem título]		MuseusIST	Jan 31, 2019
[Sem título]		MuseusIST	Jan 9, 2019

Figure IV.11: Interface to browse ISTCIDOC items.

It's also possible to edit or erase one or many items (through the protocols methods explained in Table IV.1).

3) Add New Location

 A web form titled 'Add Location' with a 'Submit' button. It shows a map of a building with a circle highlighting a specific location. Below the map, there are input fields for 'Location URI', 'Location Title', and 'Location Position'.

Figure IV.12: Interface to add new location.

4) Browse Locations

 A table titled 'Browse Locations' with a 'Add new location' button. The table has columns: 'Location URI', 'Location Title', and 'Location Position'. It shows a list of locations with their URIs, titles, and positions.

Location URI	Location Title	Location Position
2448131386442	Gabinete	Prateleira 1
2448131360897	Alameda	Posicao 1

Figure IV.13: Interface to browse locations.

Once a location is clicked, it's possible to see its details (you can also edit/erase a specific location):

 A page titled 'Arrecadação' showing details for a location. It includes a 'Location URI' (244813138670), 'Location Title' (Arrecadação), and 'Location Position' (Prateleira 1). Below this is a 3D architectural drawing of a building with a yellow dot indicating the location.

Figure IV.14: Page with details about the location.

5) Location search

 A search interface titled 'Busca avançada' with a 'Busca' button. It has a search bar with a dropdown menu showing options like '[Any Property]', '[Any Property]', 'Location Properties', 'Uri', 'Location', and 'Position'. There is also a dropdown for 'é exactamente' and a search button.

Figure IV.15: Interface to search a location.

N. Affected tables in Omeka S Database

In this case, the root table defined is called 'omeka' - this parameter is 'dbname' and is defined in the file inside the Omeka S folder in 'config/database.ini'. Also, the MySQL 'username', password and the name of the DB are also defined [29].

Here it's defined several child tables that make up all entities, such as Items, Item Set, Average, etc. Since the 'istcidoc_item' entity is defined as a subclass of the 'items' entity, then any ISTCIDOC item created by the module will be present in the 'item' table. Hence in the resource type column ('resource_type'), it appears as 'Omeka\Entity\Item'.

 A table showing an example of an ISTCIDOC item in the database. It has columns: 'id', 'owner_id', 'vocabulary_id', 'local_name', 'label', and 'comment'. The row shows 'id' 211, 'owner_id' NULL, 'vocabulary_id' 11, 'local_name' 'accessory', 'label' 'Accessory', and 'comment' 'Accessory'.

id	owner_id	vocabulary_id	local_name	label	comment
211	NULL	11	accessory	Accessory	Accessory

Figure IV.16: ISTCIDOC item example in the database.

 A table showing the 'value' table in the database. It has columns: 'id', 'resource_id', 'property_id', 'value_resource_id', 'type', 'lang', 'value', 'uri', and 'is_public'. It shows several rows of data.

id	resource_id	property_id	value_resource_id	type	lang	value	uri	is_public
15	9	1	NULL	literal		Locations	NULL	1
16	80	208	NULL	literal		11111	NULL	1
17	80	210	NULL	literal		2222	NULL	1
18	80	207	NULL	literal		33333	NULL	1
19	80	209	NULL	literal		Alameda 2	NULL	1

Figure IV.17: Table 'value' in the database.

 A table showing the 'location' table in the database. It has columns: 'id', 'uri', 'local', and 'position'. It shows two rows of data.

id	uri	local	position
4	2448131363670	Arrecadação	Prateleira 1
2	2448131386442	Gabinete	Prateleira 12

Figure IV.18: Table 'location' in the database.

V. EVALUATION AND DISCUSSION

In this chapter, we give a brief evaluation and discussion about the final product of this dissertation. This evaluation is about what has been implemented here with other alternatives and a demonstration of the created entities.

A. Evaluation of created items and locations

Omeka S has the ability for the administrator to create multiple navigation pages for his site.

The ‘Item showcase’ is the display section at the top of one or more items that you want to display, but only your title appears, you need to click on that item to learn more about it. The ‘Item with metadata’ shows one or more items below the ‘Item showcase’ with all the metadata described. In the bar on the right side, you can add several other blocks to the museum’s final presentation as Media, Tables and HTML.

You can see the details of each item too by clicking on one of them. In this case, the class to which that object belongs will already be present.

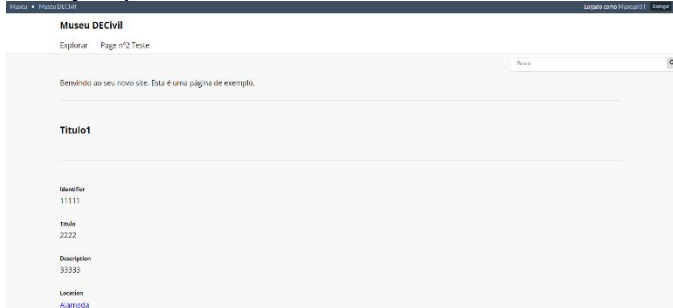


Figure V.1: Initial page of museum DECivil.

Thus, with the implementation by Omeka, you can have one or more virtual exposures, and control what you can put and not put in the final product, and what is public or private for a banal customer. The final aspect, in terms of design and aesthetics, can be changed, using various themes that are already installed in Omeka S, or by themes and/or external modules.

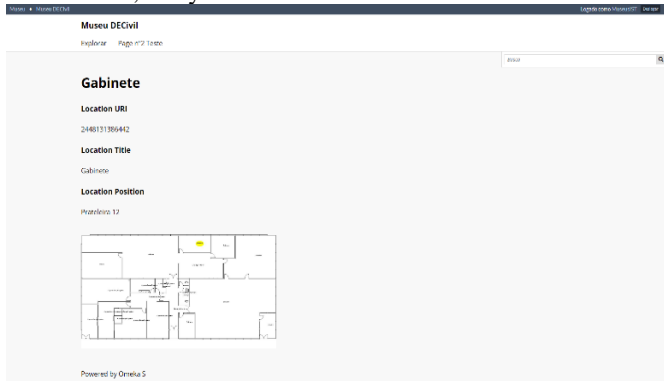


Figure V.2: Details of entity Location.

B. Comparison between Omeka and other Frameworks

This module is implemented to have an ontology of its own and well defined by the institution. In this case, it is the DECivil museum, and therefore, satisfies the basic needs of this museum, with the appropriate classes and properties, always following CIDOC rules. Hence the implementation of a new ontology, focused on the Instituto Superior Técnico’s museums.

Therefore, of all the solutions found in the different frameworks, the use of registering a module for an easy-to-work framework like Omeka S was the most adequate resolution, although it was not the easiest.

C. ISTCIDOC vs. Dublin Core

Analyzing the DC in detail, we observed a lack of properties and classes, as is the case of the ‘Location’ attribute. Therefore, CIDOC norms and parameters were used to define such classes and properties, and only those essentials were adapted to serve the requirements of the DECivil museum at an introductory stage.

In short, it was considered most appropriate to create a new ontology (ISTCIDOC) for the reasons given above, considering the possibilities of extending new attributes in the future.

D. Digital vs. Traditional Method for DECivil Museum

The only advantage of the traditional method - that is, storing objects and tokens in physical places such as shelves and shelves - is that it has something palpable and that it is preserved over time. This means that only a good storage location will have to be ensured without the concern of having to rely on the reliability and security of a digital database, which may have redundancy failures and others if insufficient care is taken.

Also, in some cases of museums, physical file support may be much more important and valuable than something digital, and it values itself immensely there.

In addition, in order to keep pace with technology and to maintain today’s (and future) modernity, it will be necessary to introduce and migrate to digital technologies, not even discarding all existing physical support (and that in most cases must always exist).

Finally, the digital method allows to attract more customers and increase the divulgation of a given museum: by means of online exhibitions and advertisements, and easy access to any user to view a museum (through a website, for example). And it is believed that for the DECivil museum (and for the other museums of Instituto Superior Técnico), this will be no exception.

VI. CONCLUSIONS

This work helped to introduce a new technology for an information system and database for the museums of the Technical Higher Institute, namely the DECivil museum (Department of Civil Engineering).

The final objective, as discussed previously, will be to integrate, at this present time, the DECivil museum in RPM. This work may not be enough to fulfil such a requirement immediately, but it is certainly a step in the right direction.

This dissertation served to remove gaps from the lack of digital support and the “sloppiness” of maintaining the traditional style for these museums. Therefore, there is no doubt that it is much easier to disseminate more (made online by Web portals and even social networks) of Instituto Superior Técnico museums.

And through this aspect, the digital standardization and database for museums, and the positive consequence that: museums become more adherent because a physical movement is not necessary for some exhibitions (the so-called virtual exhibitions), it will become more accessible for entry into the

RPM, and to make a museum with more recognition, both nationally and internationally.

A. Future work

As future work, there are some well-established topics that can be put into practice. Among them, the implementation of the same database, but instead of being done locally (as was the case for testing purposes), for cloud platforms (cloud servers)

It will also be necessary to eventually extend the ISTCIDOC ontology, and if that is the case, a module file will have to change to include more attributes, either class or property.

It is now up to those who manage the Omeka system, create and develop multiple sites - each corresponding to a museum - and add multiple pages corresponding, for example, to each permanent or fixed virtual exhibition. This also includes custom design themes for each page, since it is currently implemented with the theme from Omeka's factory, a simpler model.

Finally, a growing trend in the digital sense can be made by the efforts of the museums, in order to capture the attention of users and consumers of the entire age group.

REFERENCES

- [1] LLC, Museums and the Web, "Museums and the web," 9 Abril 2017. [Online]. Available: <https://www.museweb.net/>.
- [2] J. Kidd, "Enacting engagement online: framing social media use for the museum," *Information Technology & People*, pp. 64-77, 2011.
- [3] Stéphanie dos Santos Ramos, "Museus interativos: um novo modelo de relação?," Coimbra, 2013.
- [4] L. Gili, Use of Websites to Increase Access and Develop Audiences in Museums: Experiences in British National Museums, 2004.
- [5] C. Eades, "A Systems Framework for Museums," *Curator The Museum Journal*, 2010.
- [6] Museums, International Committee for Documentation of the International Council of, "International Guidelines for Museum Object Information: The CIDOC Information Categories," *International Committee for Documentation of the International Council of Museums*, 1995.
- [7] ICOM/CIDOC CRM Special Interest Group, "Definition of the CIDOC Conceptual Reference Model," Novembro 2011.
- [8] CIDOC, "Definition of the CIDOC Conceptual Reference Model," ICOM/CIDOC Documentation Standards Group, Novembro 2011. [Online]. Available: <http://www.cidoc-crm.org/html/5.0.4/cidoc-crm.html>.
- [9] CIDOC, "Graphical Representation of core CRM form," 21 10 2013. [Online]. Available: http://www.cidoc-crm.org/cidoc_core_graphical_representation/graphical_representation.html. [Accessed 20 10 2013].
- [10] Museu de Engenharia Civil, "MUSEU / MUSEUM," Técnico Lisboa, 2015. [Online]. Available: <http://museudec.tecnico.ulisboa.pt/museu/>. [Accessed 2019].
- [11] LIBRARIES AND CULTURAL RESOURCES, "Uploading Items in Omeka - Digital Collection Projects - Library at University of Calgary," University of Calgary, 2017. [Online]. Available: <https://library.ucalgary.ca/c.php?g=608280&p=4220142>. [Accessed 2019].
- [12] Direção-Geral do Património Cultural, "DGPC | Museus e Monumentos | Rede Portuguesa de Museus," 2017. [Online]. Available: <http://www.patrimoniocultural.gov.pt/pt/museus-e-monumentos/rede-portuguesa/>. [Accessed 2019].
- [13] C. Dan, "Introducing Omeka," 20 02 2008. [Online].
- [14] Roy Rosenzweig Center for History and New Media, "Site Planning Tips," MkDocs, 2019. [Online]. Available: https://omeka.org/classic/docs/GettingStarted/Site_Planning_Tips/.
- [15] Roy Rosenzweig Center for History and New Media, "Omeka Classic," Corporation for Digital Scholarship, 2019. [Online]. Available: <https://omeka.org/classic/>. [Accessed 2019].
- [16] Doctrine, "Doctrine Query Language - Object Relational Mapper (ORM) - Doctrine," 2019. [Online]. Available: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/reference/dql-doctrine-query-language.html>.
- [17] Roy Rosenzweig Center for History and New Media, "Vocabularies - Omeka S User Manual," MkDocs, 2019. [Online]. Available: <https://omeka.org/s/docs/user-manual/content/vocabularies/>.
- [18] J. Wiecek, D. Bloom, R. Guralnick, S. Blum, M. Döring, R. D. Giovanni, T. Robertson and D. Vieglais, "Darwin Core: An Evolving Community-developed Biodiversity Data Standard," PLoS One, 2012.
- [19] OEDb Open Education Database, "5 Free and Open Source Tools for Creating Digital Exhibitions," Accredited Online, Specialty, and Campus-Based Colleges, 2019. [Online]. Available: <https://oedb.org/ilibrarian/5-free-and-open-source-tools-for-creating-digital-exhibitions/>.
- [20] Apache Software Foundation, "About the Apache HTTP Server Project," Wayback Machine, 7 Junho 2008. [Online]. Available: https://web.archive.org/web/20080607122013/http://httpd.apache.org/ABOUT_APACHE.html. [Accessed 25 06 2008].
- [21] Roy Rosenzweig Center for History and New Media, "Users - Omeka S User Manual," MkDocs, 2019. [Online]. Available: <https://omeka.org/s/docs/user-manual/admin/users/>. [Accessed 2019].
- [22] S. Stead, "The CIDOC CRM, a Standard for the Integration of Cultural Information," CIDOC Conceptual Reference Model Special Interest Group, Crete, Greece, 2008.
- [23] The Board of Trustees of the Leland Stanford Junior University, "protégé," National Institute of General Medical Sciences, 2019. [Online]. Available: <https://protege.stanford.edu/>. [Accessed 2019].
- [24] Zend Framework, "Introduction to Zend Framework," Zend, a Rogue Wave Company, 2019. [Online]. Available: <http://framework.zend.com/manual/en/introduction.html>. [Accessed 12 02 2009].
- [25] N. Humfrey, "EasyRDF - RDF Library for PHP," BSD 3-Clause, 2019. [Online]. Available: <http://www.easyrdf.org/>. [Accessed 2019].
- [26] Governo Português, "Software livre, Boas Práticas," Instituto Superior Técnico, 06 11 2007. [Online]. Available: http://www.softwarelivre.gov.pt/boas_praticas/ensino_superior/utl/fenixedu.
- [27] W3C, "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)," 11 Dezembro 2012. [Online]. Available: https://www.w3.org/TR/owl2-syntax/#Ontology_IRI_and_Version_IRI. [Accessed 2019].
- [28] Roy Rosenzweig Center for History and New Media, "Access Control Lists - Omeka S Developer Documentation," MkDocs, 2019. [Online]. Available: https://omeka.org/s/docs/developer/key_concepts/access_control_lists/. [Accessed 2019].
- [29] Roy Rosenzweig Center for History and New Media, "Installing - Omeka S User Manual," MkDocs, 2019. [Online]. Available: <https://omeka.org/s/docs/user-manual/install/>. [Accessed 2019].