Reperio: A Collaborative Knowledge Environment for Digital Humanities

by Damiana Luzzi

Citation


Abstract

We present Reperio, a multi-user, modular, collaborative, flexible and customisable web work knowledge environment for Digital Humanities, Sciences and Cultural Heritage. Reperio has a modular and tools-based architecture. The Reperio’s architecture is composed of two modules: Ontology Editor and Text. The characteristics of Reperio will be presented on the basis of the results of the activities carried on for the management of the complex and articulated documental material owned by Historical Archives of the Pontifical Gregorian University.
Reperio: A Collaborative Knowledge Environment for Digital Humanities

by Damiana Luzzi

1. Introduction

The characteristics of Reperio, a multi-user, modular, collaborative, flexible and customisable web work knowledge environment for Digital Humanities, Sciences and Cultural Heritage, will be presented on the basis of the results of the activities carried on for the management of the complex and articulated documental material owned by Historical Archives of the Pontifical Gregorian University (APUG). APUG, which is unique in its own right, testifies the intellectual activity of the Jesuits of the Roman College, from its foundation in 1551 to the suppression of the Society of Jesus in 1773. The Roman College was later reopened in 1824, ten years after the restoration of the Society, and continues its educational function as Pontifical Gregorian University. No other university can show a continuous record of teaching for nearly five centuries. Because of this long history APUG describes, probably better than any other European institution, the educational, teaching and research practices over a long lapse of time. Its funds cover all the subjects taught in the Roman College such as astronomy, grammar, mathematics, philosophy, rhetoric, theology, etc., and studies on Latin and Greek classics, all of them preserved as texts used personally by the teachers during their lessons. Moreover, its funds attest to the extra educational activity of the teachers, especially for the XIX and XX centuries (e.g. hold documentation - codices and newspapers - from the Council of Trent and from Second Vatican Council, where some Jesuits professors at Pontifical Gregorian University had an important role). At the end, the troubled vicissitudes lived, especially in Rome, by the Jesuits and other religious Orders’ goods after the unification of Italy, have brought among the APUG funds a lot of “stranger” documents, sometimes totally disconnected from its history.

Because of such history, APUG owns diversified material. It comprises more than 5000 codes attesting the lessons of rhetoric, grammar, philosophy and theology, that were hold during two centuries, in addition to the studies of Greek and Latin classics, astronomy, mathematics and physics, and Latin, Hebrew, Greek and Arabic languages. Along with this material, other
important documents attest to the fervid activity of research and study that took place at the Roman College:

- The correspondence of Athanasius Kircher.
- The correspondence of Christopher Clavius.
- The codes used by Sforza Pallavicini to write the Story of the Council of Trent.
- Other documents show the relations that many Jesuits around the world maintained with the masters of the Roman College.
- The first topographic maps of China made by the Jesuit D'Elia and documents on the missions in Asia.

The material includes:

- Printed texts, some of these glossed by author and with notes of the censors.
- About 6,000 manuscripts with glosses, erasures and some of these with insertion of papers and fragments.
- Many modern archival documents.
- Ancient and modern books, both with a lot of handwritten notes.
- Graphic material, such as drawings, prints, maps and photos.

This variety of resources implies at least two kinds of problems connected with:

1. Preservation: each kind of resource has its own preservation requirements.
2. Cataloguing: to describe different resources we should use the standards developed for each of them: ISAD(G)\(^6\) for archival documents, FRBR\(^7\) for bibliographic material or ISAAR and FRAD\(^8\) for authority records, and so on. Usually these standards have their own software expressly developed to simplify their use and application.

However, this situation is common not only in many archives, but also in libraries and museums.

**2. Reperio in APUG**

Trying to give a different answer to the second of these problems APUG, in 2010, begin a close partnership with Fondazione Rinascimento Digitale (Digital Renaissance Foundation, FRD) – an institution specialised in the
development of most advanced solutions for cultural heritage – and the Institute of Computational Linguistics “Antonio Zampolli” (ILC, an institute of the National Research Council of Italy). Thanks to this partnership, the archival and cataloguing competencies owned by APUG have been turned into a more technological output by FRD with Reperio.

This is an exciting challenge which involved Reperio on two levels:

1. To represent the complex physical, structural and textual description of the resources, often printed-manuscript “hybrids” (Figure 1), whose levels of description are at least four: published text, glosses, later interventions, correspondence; and to collect at the same time data related to the binding, the state of conservation and restoration of the documents.

2. To reconstruct, starting from the use of manuscript and later elaborations, the history of education and its evolution within the Roman College.

Figure 1: Cinquecentina with glosses and interlinear.

APUG 429. Biblia ad vetustissima exemplaria nunc recens castigata. Hebræa, Chaldæa, Græca et Latina nomina virorum, mulierum, populorum,
idolorum, urbium, fluviorum, montium, caeterorumque locorum, quae in Bibliis leguntur, restituta cum Latina interpretatione, ac locorum e cosmographis descriptione, Louanii, ex officina Bartholomaei Grauii, 1547.

To manage these levels we developed an ontology,⁹ which could solve the cited cataloguing problems without neglecting the principles set by the international federations of libraries and archives (IFLA¹⁰ and ICA,¹¹) especially in terms of standardisation and conceptualisation,¹² and in order to make available an open and flexible system in which information, with a different granularity, can be integrated and continuously updated by the users themselves.

3. Reperio’s Architecture

Reperio reverses the perspective: it brings the user to the forefront and places technology at his service. Therefore, the strategy supported by Reperio is a bottom up approach. Specialised, flexible and reusable tools are implemented on a modular architecture, which provides the core functionality. These tools are developed to meet the needs of the projects¹³ thanks to an exchange of information with users too.

The Reperio’s architecture is composed of two modules:

1. **Ontology Editor** allows editing of classes and properties and the instantiation of the ontology
2. **Text** allows the editing and the management of texts and/or images. It provides specialised tools that offer various functionalities.

These two modules can work together or separately.¹⁴ It is possible to work collaboratively in every Reperio module because they can be accessed via web using a computer with a web browser and an internet connection (internet cable, wifi, internet key or tethering), and therefore without any barrier with respect to location.

4. Reperio Ontology Editor

We used the **Reperio Ontology Editor** to develop the ontology and instantiate it. **Ontology Editor** is divided in two main functional area:

- **Ontology Management**: allows the creation, modification and updating of
ontological schemas (classes, properties and restrictions on them). It also contains parameters to configure the Data Input (Module 2), which define the types of fields (text boxes, multiple or single choice drop-down menus) and orders them.

- **Data Input**: is used to insert data (instances) in the ontological schema, to carry out their editing and to research them in order to verify their accuracy.

*Ontology Management* and *Data Input* communicate in a dynamic way: any changes made to the ontology schema, such as the addition, modification, cancellation of a class or a predicate, is instantly transferred to the *Data Input*. You do not need the assistance of a programmer to change the parameters of the *Data Input*, and even ontological schema changes can be made by an experienced, well trained and authorised user.

### 4.1. Ontology Management

Now, we see the interface of the *Ontology Management* (Figure 2). On the left we have the ontology schema, on the right: at the top we have the information about the class, and at the bottom the information about predicates or relationships between *Domain Class* and *Rank Class*.

The operations that can be performed on classes are run from the drop down menu in the footer of the page. The operations are:

- **Extend** or specialised class: means that to a new class can be added as subclass and their definition at the attribute level can be customised.
- **Manage** class: edit, change and add properties of the chosen class.
- Add a binary predicate: add a predicate between a domain class and a rank class - e.g. a *Expression* (domain class) *isEmbodiedIn* (predicate) a *Manifestation* (rank class) where the domain is the subject and the rank is the complement.
- More information about the class.
- Check the labels of all classes.
- Check the labels of binary predicates.

You select the task of interest and click on Execute to open the pop-up window.

For example, when you insert a new class you position the mouse cursor at the point of the scheme where you want to extend (add) a new class and then select Extend in the drop down menu, you click on Execute to open the pop-up window, and then you insert the class name; if the class is abstract
(the abstract class can not have direct instances); if the class is mandatory; if the class can be specialised (final) or not (if we select yes, the class will not have subclasses); if the class is type, it means it is a service class; if the value of the class is scalar, that means that the values are alphanumeric string or not; and finally if we want to describe the class content.

Figure 2: Ontology management.

4.2. Example: The Actor Class and its Specialisations

The organisation of information about “actors” is inspired by FOAF\textsuperscript{15} ontology: opting for Actor (Figure 3) instead of Agent, more familiar term in the humanities and more generic of the term Creator present in ISAD (G). The Actor class is specialized in classes Group and Person. The subclasses of the Group are:

- Organisation (e.g. institutions, organisations, agencies, public or private companies) specialised in classes: Family, ReligiousOrder).
- TemporaryOrganisation describes bodies of a temporary nature (e.g. committee, consortium, conference).

The Person class is related (isMemberOf) with to Group this relationship indicates whether one or more persons are part of a group. The classes Person and Group have other relationships and attributes respectively to enter:
• Date and place of birth and death.
• Date and place of foundation and termination of activities.

On the right, in the interface of the Ontology Management (Figure 2), we can see the predicates that define the Person class. The predicates with the gray background are those inherited from the superclass, and in white the specific predicates of the Person class.

In cataloguing systems it is very important how the names are structured: a simplification could affect the generation of the relevant indices. Appellation class is used to refer to and define the “names” of: people (PersonName), groups (GroupName), places (PlaceName), objects (Name) and titles (Title). The Names are placed in the normalised form indicating, where applicable, the authority file reference (e.g. VIAF) and in their possible variant forms (e.g. Clavius, Cristoforo Clavio, Christophorus Clavius, Christoph Clavius, Christoph Clau) also indicating the time and place in which the variant is in use (e.g. some Jesuits have lived in China and they have changed their name).

In addition, the classes responsible for the management of the names have been designed both to allow simple insertion, following the rule of the inclusion of cataloguing national or international. So, the first name, last name and the particle name are entered separately in order to decide how to build the index of names and how to display the name of the person in the search interface: first name and last name, last name followed by a comma and then the first name, etc.

Instead of creating many classes to describe the relationships between actors (Actor) and objects (Object), we have created the only Role class. Role, linked to ActorInRole, allows us to express, without predetermined patterns, each possible role that an actor can take in the function of an object (eg, author, publisher, sender, recipient).

ActorInRole, ActorInQualification and ActorInPlace, with their relationships and attributes, to structure biographical information of persons or institutions offering also the environmental context.

Label allows you to translate in one of five languages - English, French, German, Italian and Spanish - the names of classes and predicates that we use in Data Input (Figure 3).
4.3. Data Input

Let us see now, how the Data Input views our classes and predicates (Figure 3). First of all the drop down menu shows us the class list. It is the list of all the classes that we have defined on the schema. We choose one of them and start from there to add or edit our data. Example of working: data entry of a person. You choose the Person and click on button New to open a new form to insert the data about a person following the order and type that we have decided in the ontology schema. If you want to change the data already entered use the box filter or the Advanced Search.

Figure 3: Data input.
The Data Input interface uses the rules defined on the schema to determine which kind of functions should be assigned to each field (pull down menu, simple field, etc.). This dynamic method to generate the interface gives the user a chance to model their own data also during the research activity. Every change that has been made in the Ontology Management is
immediately transposed in *Data Input* too.

## 5. Reperio Text

And now, after having seen the tool we use to catalogue the resources, we are going to describe how to manage the text transcription and the image on *Reperio Text*. *Text* allows the editing and the management of texts and/or images. It provides several tools that offer various functionalities, for example:

- Text editor.
- Digital image manager.
- Metadata, full-text and semantic search.
- Annotation editor.
- Comparison manager.

### 5.1. Access

The access to *Text* module is carried out through a procedure of authentication (login). *Text* can be used by the user Guest (unauthenticated) but in this case the number of available features is reduced. The following roles anticipated by current user profiling include: administrator, editor and player. New profiles (e.g. annotator and more) can be defined and added at any time.

Within the collaborative environment, each category of users is provided with tools, services and an interface specialised in the operations that shall be carried out.

### 5.2. Interface

The Reperio *Text* interface (Figure 4) has a layout determined that involves four fundamental areas. At the top there is a *Menu bar* for quick navigation of the pages in the application. The central part of the page is divided into two parts: the Toolbar on the left and the *Main content* on the right. The *Toolbar* displays the tools that are available. The *Toolbar* is resizable. In addition, at any time, you can hide the *Toolbar* to work in full screen or to re-establish the hidden toolbar.
The number of operations you can perform on a document will grow. In order to realise a modular and expandable interface we carry out various functionalities, such as panels of separate tools. In this way the addition of new features corresponds to the addition of new specialised panels. The Main content presents the main contents of the currently displayed page. The last of the four areas is the Footer, on the bottom.

5.3. Search

It is important to promote the “findability” of resources to allow the user to find what he is looking for as quickly as possible and with as little “noise” as possible, to have different ways of searching from the simpler “googlelike” style to the most advanced one, to indexes and pathways. With the tool or (plugin) Filter Documents, you can choose among the various search options:

- For metadata: title, author, date, etc. This metadata can be added or deleted according to the needs of the project.
- Content.
- Semantic.

These options can be combined.

For example you can do a query that will search for documents whose content has at least one word between “light and dark and contains the words grandezze and bilancie a distance of 5 and does not contain the word calendario”. You compose the search expression manually: `(( grandezze && bilancie ) && (( dimostratione DIST#5 teorema ) && calendario ))`. 

---

**Figure 4: Reperio text interface.**
The query tool **Composer** (Figure 5) offers an easier way to write complex queries to search in the content of the text. The expression of the search can be as complicated as preferred. You can also associate a search by metadata (e.g. author: writing “Galileo” in the field **Authors**). When you search for content, for each document found is shown a list of text extracts that have determined the success of the research (Figure 6). On the right are shown the results of the research. The results can be sorted by Title, Author and Date. The View button displays the document.

**Figure 6: Search results.**

5.4. Document

The interface for the visualisation and manipulation of documents allows you to choose between various viewing modes (only transcription, transcription and image, etc.).

The creation of a new document (text and/or image) is performed manually
or through automatic import. There is a WYSIWYG (an acronym for the English phrase: What You See Is What You Get) text editor, that is very useful in the transcription of a text.

**Figure 7: Text editor and image selection.**

In the text editor there is an interesting function: *Image selection*. *Image selection* allows you to select one or more portions of the image, copy and insert them (e.g. mathematical formulas, graphs, drawings) in the text (Figure 7).

You can assign permissions (read, editing, annotation) on the single document too.

5.5. Available Tools

Well, the document searched for has been accessed, the *palette* of available tools changes and becomes enriched:

- **Document search**: for searching within the document content;
- **Metadata Document**: to view and edit metadata about the document such as Title, Author, Editor, Date, etc. You can enter information on one or more authors. Note that this tool is easily extendable to contain new metadata such as variants of the title, one or more editors.
- **Metadata Page**: to insert the page number of the original page. This tool is easily extendable to contain new metadata related to a single page, such as
information on the conservation status.

- **Index**: shows the logical structure of the document. It is possible to jump to a particular section of the document by clicking on the corresponding link. You can edit the logical structure of the document by going into the *Edit document* and selecting the tag Heading (1, 2, 3, 4, 5, 6) in the WYSIWYG text editor.

- **Index of term**: shows the index of the terms in the current document. It sorts the list and searches for a term. It is possible to jump to a particular section of the document that contains the term: you just have to click on the corresponding link and go to term (you write the term). Alphabetical order (direct, inverse) or order for frequency number.

- **Pages**: shows the physical structure of the document, you can see all the pages of a document in an icon format. This tool allows you to navigate between the pages of a document and allows you to add, edit, change the order or delete pages.

5.6. Annotation

The annotation is “the act of taking note”, writing comments, posting links to other resources, inputting metadata, tags, etc. The annotations achieve their full significance in relation to the resource “target” (text and/or support) and other contextual information such as its author, creation date and vocabulary of terms used.

A look at the model. For this description we use the standard language UML (Unified Modeling Language).

*Figure 8: Annotation (UML schema).*
Annotation\textsuperscript{17} (Figure 8) is the central class put in relation with:

- **Actor** indicates who performs the annotation and by whom it can be seen. It specialises in subclasses **Machine** (annotation accomplished automatically by a software agent) and **User** (subclass of **Person**, annotation accomplished by a person or user of the web application).
- **Date** detects the time at which the annotation was made.
- **AnnotationType** provides information about the type of annotation (eg, comments, links, tags, semantic annotation, linguistic annotation, morphological annotation) in association also to the vocabulary (**Vocabulary**) in use.
- **Vocabulary** specifically glossary, thesaurus, vocabulary, ontology used for annotation. It draws attention to the classes described above, **Object**, **ConceptualObject**, **PhysicalObject**, **ItemStructure**, **ActorInRole**, **Actor**, **Material**, **Damage** and because of their specialisation are used to record and collect information on the text, content, and support.
- **Selection** identifies the selection or selections of text and/or image object of the record; has subclasses **Text**, **DigitalObject**, **Image**, and any additional specialisations. The reports **hasBody** and **hasTarget** combine two different pieces of information recorded.

Comments are the most simple type of annotation possible. It allows you to post the comments on text and/or an image, and, after having posted them, to run a search both on the text referred to in the comment and on the comment itself. If the comment was made only on the image, the search takes place by indicating the coordinates of the image portion that was the
object of the comment. This feature is also present in the OntologyAnnotationTool.

In this context we present the collaborative Ontology Annotation Tool of Reperio, because it is the most widely used tool for the studies of APUG materials. The annotation is a core practice to scholars too.18 The expressive power hidden in the texts can be further maximised by combining ontology and annotation: annotation expresses and encodes, in a formal manner, the meaning of a text using the “terminology” provided by the ontology. Thanks to this type of annotation, users can study and analyse the text in different ways: philological, syntactic, morphological, grammatical, etc. They can even comment on physical materials (inks, colours, damage, watermark) or about person, place, scientific instrument, theory, etc.

Particularly, the semantic annotation19 helps to bridge the ambiguity of natural language in expressing notions and their computational representation in a formal language.

The annotation operation can be performed:

1. Through concepts associating classes to the selected term/s. If the class relating to the concept on the text is not in the ontology, it can be easily inserted (by users with appropriate permission) opening the ontology editor. For example: in the text and/or image selected, the term luna is connected to the AstronomicalObject class of the ontology.  
2. Through instances.  
3. Directly populating the ontology: selecting “astrolabio” and inserting it as instance of the ScientificInstrument class.

The instances are already in the ontology (Figure 9). In this way, the annotated element acquires the property values that describe the instance. For example: Galileo Galilei, in the letter20 that he wrote to Christophorus Clavius, he never mentions explicitly Christophorus Clavius, but he refers to him by the acronym VSMR. So, if we execute a full-text search looking Christophorus Clavius??, Galileo’s letter is not returned in the results of the research. So, thanks to Ontology Annotation Tool we can select the acronym VSMR and connect it to the instance Clavius of the Person class of the ontology. The instance Clavius can have the properties: place and date of birth, place and date of death, texts, variants of the name in other languages, roles, etc. that add more information to the acronym VSMR noted in the text (e.g. search for “Clavius” shows the results where Clavius is quoted “VSMR” too, and a search for person shows Clavius in the result).
Another example: Ilario Altobelli writes « Et si concidit cu mea forma quae exhibetur in Astrolabio. pag.. 261. 22 prob. et in exemplo pag. 703 »²¹; by semantically linking Altobelli’s citation with the instance Astrolabio²² of the Title class of the ontology, you get the reference to Christophorus Clavius's text and its bibliographic information. Therefore a portion of the ontological schema is based on FRBR-00 standard. In addition, if Christophorus Clavius’s book is present in a digital edition in Reperio or in another digital library, through a URI connection you could read the page to which Altobelli refers.

Annotation operations and/or the ontology population may be performed manually or in a semi-automatic manner by text parsing performed on the basis of the classes and/or instances.

The semantisation of knowledge significantly improves accuracy and relevance of search results.

The OntologyAnnotation tool also allows you to write additional information on the annotation, and then to perform a search on them too.

Figure 9: Semantic annotation.

6. Conclusion

Ontology and a collaborative environment like Reperio can be considered an evolving open system, that offers scholars the freedom to work, search and explore. On the experience of this and other projects, the process of designing an ontology has been very useful, because it offers different views
and perspectives on texts and the concepts that they convey, and will open new ways for further studies and analysis. Such an “enhanced” search allows you to infer and deduce new knowledge based on what is available.

We are implementing other tools and functionality to allow Reperio to respond better to the needs of our users:

- Linked Data: Digital resources are also identified by the Uniform Resource Identifier (URI), the same identifier at the base of Linked Data.\(^\text{23}\) It is useful to provide the URI the digital resources to put into Reperio to bring together information about digital resources produced by other projects or national and international research centres, other archives and libraries and also, if you wish, make available the resources (all or in part) produced by the project which uses Reperio.
- To facilitate sharing, interoperability and reuse of information we are studying a tool that allows Reperio to exchange information with Europeana.\(^\text{24}\)
- To map the ontology schema developed with Reperio to transfer its data to the Europeana Data Model;\(^\text{25}\)
- To implement procedures for metadata harvesting and exchange based on OAI-PMH\(^\text{26}\) and OAI-ORE.\(^\text{27}\)
Footnotes

1 Developed by the FRD in collaboration with the Institute of Computational Linguistics “Antonio Zampolli” (ILC-CNR), http://www.reperio.it. Reperio takes advantage of experience gained during the experimental project Pinakes of which it is an evolution. The Pinakes project was the result of a long-term activity within the research framework of the Museo Galileo (ex Istituto e Museo di Storia della Scienza) in Florence. Pinakes 3.0, the new version of Pinakes was developed from 2004 to 2011 by the Fondazione Rinascimento Digitale (FRD, Digital Renaissance Foundation, http://www.rinascimento-digitale.it, and the Institute of Computational Linguistics “Antonio Zampolli” (ILC) of CNR, in collaboration with the Ministry of Cultural Heritage and Cultural Activities and the Galileo Museum.

2 Historical Archives of the Pontifical Gregorian University (APUG), http://www.archiviopug.org.

3 When the Society of Jesus was suppressed, the archives were hidden (buried and walled) for safekeeping. One hundred years later, it was discovered by the librarian of the National Library in Rome. Currently, half of them are in the National Central Library of Rome and the rest at the APUG.


The ontology was developed by Damiana Luzzi (FRD) and Irene Pedretti (APUG).


They are installed on two different virtual machines. If a project regards only the cataloguing it uses module one (Ontology Editor), if the project concerns the view, editing, search, etc. of the text it uses module two (Text). But if at any time the project needs to use the other module too, this can be installed and used immediately. The Reperio Text is connected to the ontological module that uses for the different types of annotations that are based on the ontological schema.


Virtual International Authority File (VIAF), http://viaf.org.

In developing the portion of the ontology annotation we have considered the specification of the Open Annotation Core Data Model although the
document is in draft status. OAC data model provides a description of a standard model for the sharing of records among systems. W3C, Open Annotation Collaboration, Open Annotation Core Data Model. Community Draft, February 8 2013, http://www.openannotation.org/spec/core.


20 Galileo Galilei to Christophorus Clavius, January 8 1587.

21 Ilario Altobelli to Christoph Clavius in Rome , Montecchio June 23 1604, APUG 529 cc. 247r-248v.

22 Christophorus Clavius, Astrolabio, APUG 775;manuscript consist of three books; imp. Bartholomaei Grassi, ex tyographia Gabiana (Rome)1593.


24 Europeana is an internet portal that acts as an interface to millions of books, paintings, films, museum objects and archival records that have been digitised throughout Europe, http://www.europeana.eu.

