

# ONTO-H: A collaborative semiautomatic annotation tool

Benjamins V.R.<sup>1</sup>, Contreras J.<sup>1</sup>, Blázquez, M.<sup>1</sup>, Niño M.<sup>1</sup>, García A.<sup>2</sup>, Navas E.<sup>2</sup>, Rodríguez J.<sup>2</sup>, Wert C.<sup>2</sup>, Millán R.<sup>3</sup>, Doderó J.M.<sup>3</sup>

<sup>1</sup>Intelligent Software Components, S.A. [www.isoco.com](http://www.isoco.com)  
{rbenjamins, jcontreras, mercedes} @ isoco.com

<sup>2</sup>Residencia de Estudiantes [www.residencia.csic.es](http://www.residencia.csic.es) Spain

<sup>3</sup>Texto Digital S.L. Spain

<sup>4</sup>Universidad Carlos III <http://www.uc3m.es/>  
doderó@inf.uc3m.es

## Abstract

Online cultural archives represent vast amounts of interesting and useful information. During the last decades huge amounts of literature works have been scanned to provide better access to Humanities researchers and teachers. Was the problem 20 years ago one of scarceness of information (precious originals only to consult in major libraries), today's problem is that of information overload: many databases online and many CD collections are available, each with their own search forms and attributes. This makes it cumbersome for users to find relevant information. In this paper, we describe a case study of how Semantic Web Technologies can be used to disclose cultural heritage information in a scalable way. We present an ontology of Humanities and a semi-automatic collaborative tool for annotation built as a plug-in of Protégé 3.0 and using the Protégé 3.0 server.

## Introduction

Online cultural archives represent vast amounts of interesting and useful information. During the last decades huge amounts of literature works have been scanned to provide better access to Humanities researchers and teachers. Most works are scanned as images, which are available through microfiches, CDs and online databases. More recently, OCR techniques are increasingly applied to provide full text search facilities. Was the problem 20 years ago one of scarceness of information (precious originals only to consult in major libraries), today's problem is that of information overload: many databases online and many CD collections are available, each with their own search forms and attributes. As is the case with general search engines, keyword based search has its limitations. One can only search for specific words and their cooccurrence in documents. This may give acceptable results for 'normal' Web users, in the area of humanities it is not sufficient because one is above all interested in relations e.g. between artists, their works, the friends, their studies, who they inspired, etc. In this paper, we describe some of the products obtained as a result of projects ESPERONTO, ONTO-H and SEGEPAC can be used to disclose cultural heritage information in a scalable way. We present an ontology of Humanities and a semi-automatic tool for annotation. The basic idea is the following:

- Build an acceptable ontology of Humanities by involving professionals.
- Use the ontology to semantically annotate existing cultural content.
- Support the annotation process by an "intelligent" editor.
- Provide a collaborative environment.

## An Ontology of Humanities

To build the ontology, the Competency Questions Methodology [6] has been used. Some of the concepts included in the ontology are: Studies, Profession, Company, Institution, Academic Organization, Person, Work, Expression, Manifestation, Graphic works, Literary work, Musical work, Event, Exposition, Group, Social Group, Movement, Subject. A complete description of the ontology can be obtained from [1]. Each concept of the ontology is described through several attributes. The ontology is written using RDF language.

The ontology and the first version of the intelligent editor have been initially constructed with Protégé 2000. The intelligent editor first version has been migrated to Protégé 3.0 [9].

## The Semi-Automatic Annotation Tool

The annotation task for the Semantic Web takes as input existing content, either structured, semi-structured or unstructured, and provides as output the same content along with a semantic annotation

based on ontologies. The semantics as such are defined in ontologies. The annotations provide pointers to these ontologies.

Annotation can be performed in several manners, ranging from completely manual to tool-assisted to fully automatic. As a result of the analysis performed in [1], it turns out that the type of annotation approach to be chosen depends on the rate of structure the content exhibits. More structure allows for more automation, while maintaining the quality of the annotations.

As has been the experience of several researchers and practitioners, the annotation effort is a serious barrier to its widespread use [1, 11, 11]. Although in the area of humanities manual annotation efforts are considered as necessary and are actually performed, significant improvements are possible ranging from intelligent assistants to (semi)-automatic annotators.

Together with a detailed methodology for ontology management and population we built an intelligent annotation editor. There are considered two kinds of users for the tool:

- Knowledge engineer and reviewer (in charge of ontology schema management): performs major changes on the ontology, especially on the ontology schema, evaluating the final impact on the existing instances, and approving or rejecting the information added by the annotators at the ontology.
- Annotator (in charge of ontology population): introduces new instances in the ontology and maintain the existing ones.

The annotation tool, a kind of intelligent editor that helps the user to perform annotation tasks for a given source text is developed as a plug-in to the Protégé 3.0 tool. As shown in **¡Error! No se encuentra el origen de la referencia.**, the editor allows loading the source text to be annotated (right hand side). It allows standard editing operations on the ontology and the instances as provided by Protégé 3.0 (left hand side). Apart from the standard Protégé functionalities, the user can easily add new instances using a drag-and-drop facility.

For instance, we can say that Picasso is an artist by selecting "Picasso" in the source text and dragging it to the ontology concept "Person" and releasing it. This creates the instance and pops up the Protégé form for creating instances. The annotation process does not change the source text itself (i.e. by putting the ontological tags in the text), rather it creates a link from the instance to the original string in the source text that caused its creation. As such the original string is an occurrence of the instance, and there can be many occurrences of the same instance throughout the text. Thus, if we drag-and-drop the identical string "Picasso" again on "person", then rather than creating a new instance, it creates a new occurrence of the same instance, unless, of course, the user decides otherwise. We can also select a string like "inspired-by" and drag-and-drop it onto the corresponding relation in the ontology (modelled as a class). Now the editor creates an instance of this relationship and pops up the corresponding Protégé 3.0 form where the user is prompted among other to complete the domain and range of the relation.

Since complete manual annotation is a tedious and error-prone work, we added improvements towards automatic annotation to the editor. Automatic (cascading) creation of instances and a recommendation facility, each of them discussed in more detail below.

### **Annotation Rules**

In complex domains as described here, when performing massive knowledge acquisition tasks, there often exist typical annotation patterns to be performed. For instance, each time a new artistic work is being annotated, it makes sense to also create new instances for its expression and manifestation (as defined in the IFLA standard [6]). In IFLA "work" is defined as the idea of an art work, like that of the Guernika painting of Picasso (Spanish Civil War and the Guernika bombing). The "expression" is a painting (it could as well be expressed in a poem or movie). The "manifestation" is the actual painting that can be enjoyed in the "Reina Sophia" museum in Madrid, Spain. This kind of patterns stem from dependency relations between concepts in the domain ontology.

For these typical annotation sequences we included a rule engine that allows conditional firing of a rule set to add a new instance, check for name conflicts or consult the annotator for ambiguity resolution, among others. The rule engine is based on the open source software Drools [4] and was connected to the cultural domain ontology using an ad-hoc programmed java proxy.

### **Recommendations**

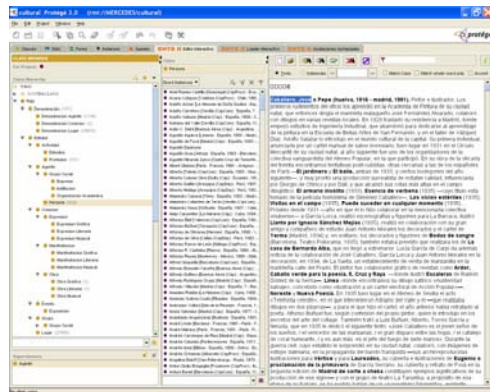
In order to increase the accuracy and speed of the annotation process the editor includes recommendation functionalities for the annotator. When the users asks for advice for selected words or text parts the systems first checks existing instances in the domain ontology. The check is performed using a Natural Language Processing (NLP) module that decides whether two instances could be the same. This subsystem includes simple lexical and morphological modules possibly augmented with a synonym dictionary. If the selected word or a part of the selected text is identified as a possible new occurrence of

an existing instance the system asks the annotator to decide which of the following action should be performed: (i) adding a completely new instance, (ii) modifying on existing instance with a new occurrence (adding new source link) or (iii) discarding any ontology modification. The more instances the ontology contains, the better recommendation the system can offer.

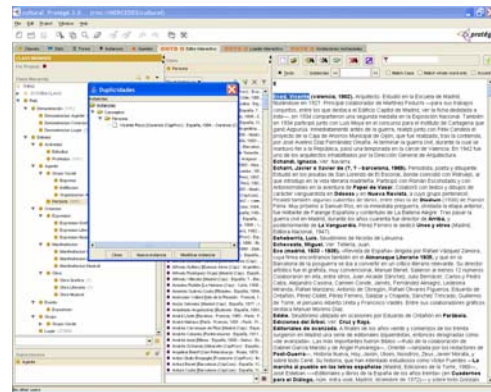
Another way of advising is firing guessing rules. These rules can suggest possible concepts for the selected text. For instance if a selected word starts with capital letter, it is not the first sentence word, and it was not recognized in the morphological module, it will be designated as a proper name and all the concepts “NAME” (person’s or place’s) will be suggested to the annotator.

### Conflict Resolution

One of the most complex concepts in this cultural ontology is NAME. Since almost all things can be named in different ways and the exploitation will stress access to the information using this attribute we took special care modelling it. Each author, place, work, etc. can possess a number of names, variable in the time line or different depending on the relation they participate in. For instance, an author can write a book using one pseudonym, then use his legal name when attending an exposition and use an acronym when writing a new book with two colleges. All these names should point to the same person instance. That is why the system offers instance duplication detection that warns the annotator of a possible existing instances using the name relations between concepts (see Figure 2).



**Figure 1:** Assisted Editor for Cultural Text Annotations



**Figure 2:** Instance duplication

### Search Facilities

We tested our tool by several users. It turned out that search facilities are extremely important in order for the annotator to keep track of what has already been annotated, as well as allowing him to follow different annotation strategies. Two example strategies include: i) following the text from begin to end, and ii) topic (author, work, movement) based. Especially in the latter strategy, search facilities are extremely important. The editor allows the following types of search:

- Marking instance from ontology: Instances already annotated have link to text marking their occurrences. It is very useful for the annotator to check what text was already processed.
- Search instance in ontology: This functionality is often used as part of the recommendations. The user can search for ontology instances that contain part of the text in their source.

All search functionalities take into account that almost all instances are referenced with their ‘name’ attribute. That is why when searching for occurrences the system identify instances not only by source but also by their ‘name’ value related.

### Import Facilities

Since there exists a bibliographic standard for authors and works storage defined upon the XML language called MARC [7], we built an import tool for translating it into the ontology formalism. Besides, this tool imports data from XML files with a specific structure, containing persons, places and their names, activities, relations between person and activities, relations between places and relations between persons and places. Some of these data comes from the CINDOC [2]. The tool uses the rule engine and parsers to process XML input file and fill the ontology with acquired instances for persons, and artistic works. As well as the drag and drop functionalities this one also includes conflict detection for avoiding data duplication. Whenever the import tool detects possible data repetition, it postpones the decision to the end of the process and then asks the user to resolve possible actions to be taken: (i) adding new instance, (ii) adding new occurrence of an existing instance or (iii) skipping any action.

### Collaborative tool.

Also as a result of the project SEGEPAC [10], ONTO-H has been migrated to a collaborative environment, which aim is to provide a collaborative annotation tool that allows expert users to build and fill the ontology in a collaborative and consistent manner. This collaborative tool has been implemented using the Protégé 3.0 server and building a new plugin that supports a collaborative protocol with two main user roles: reviewer and annotator, described below:

- The reviewer reviews all the ontology modifications made by the annotator, considering that all of these modifications made up by the annotator during a working session constitute a package. If the reviewer rejects a single modification of an instance or a new instance, he rejects all the modifications included at the package. The same occurs if a reviewer accepts a package: he/she will accept all the instance modifications made by the annotator during a working session. Finally the reviewer can modify the contents of an instance and work with the editor as if he/she were an annotator; that is, he/she can use all the operations described above.
- The annotator is in charge of the ontology population task, using all the functionalities described in this paper. In the collaborative context, the annotator will receive messages, accompanied by comments, if there is any package rejected by the reviewer. The annotator can read the comments made by the reviewer and modify the package content (adding new instances or modifying the existing ones). These modifications result in a package state change, which is pending to review again.

The annotation tool integrates specific interfaces to support this collaborative protocol, as we can see at Figure 3 and Figure 4.

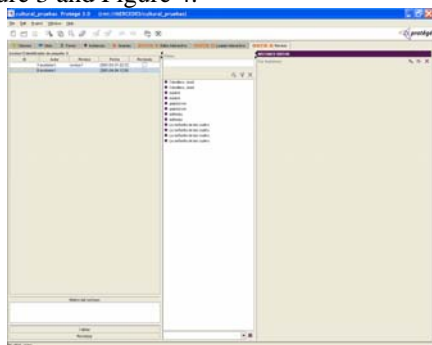


Figure 3: Reviewer interface

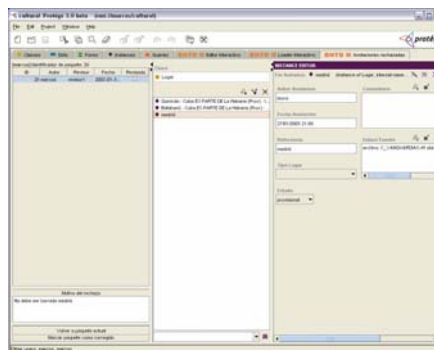


Figure 4: Annotator interface

### Acknowledgements

Part of this work has been funded by the European Commission in the context of the project Esperonto Services IST-2001-34373 and SWWS IST-2001-37134 and by the Spanish government in the scope of the projects ONTO-H (PROFIT, TIC): Intelligent access to digital cultural content based on an ontology for Humanities and SEGEPAC (PROFIT): Semantic Services for the Management of the Cultural Heritage.

### References

1. V. R. Benjamins and D. Fensel. Editorial: Problem-solving methods. *International Journal of Human-Computer Studies*, 49(4):305–313, October 1998. Special issue on Problem-Solving Methods.
2. CINDOC <http://www.cindoc.csic.es>
3. Contreras et al. D31: Annotation Tools and Services, Esperonto Project: [www.esperonto.net](http://www.esperonto.net)
4. Drools <http://drools.org/>
5. Juan Manuel Dodero, Jesús Contreras, Richard Benjamins, Test Case Ontology Specification Cultural Tour. D9.2, Esperonto Project, [www.esperonto.net](http://www.esperonto.net).
6. Federation of Library Associations and Institutions: <http://www.ifla.org>
7. MARC <http://www.loc.gov/marc/>
8. M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2): 93-155, 1996
9. Protégé tool <http://protege.stanford.edu/>
10. SEGEPAC Servicios Semánticos para la Gestión del Patrimonio Cultural. PROFIT 2004.
11. W. Swartout and A. Tate. Coming to terms with ontologies. *IEEE Intelligent Systems and Their Applications*, 14(1):19–19, January/February 1999.
12. D. A. Waterman F. Hayes-Roth, D. B. Lenat. *Building Expert Systems*. Addison Wesley, 1983.