

# Eikonomia – An Integrated Semantically Aware Tool for Description and Retrieval of Byzantine Art Information

Konstantinos N. Vavliakis<sup>1</sup>, Andreas L. Symeonidis<sup>1,2</sup>, Georgios Th. Karagiannis<sup>3</sup>,  
Pericles A. Mitkas<sup>1,2</sup>

<sup>1</sup>*Electrical and Computer Engineering Dept.,*

*Aristotle University of Thessaloniki, GR541 24, Thessaloniki, Greece*

<sup>2</sup>*Intelligent Systems and Software Engineering Laboratory,*

*Informatics and Telematics Institute – CERTH, GR570 01, Thessaloniki, Greece*

<sup>3</sup>*“ORMYLIA” Art Diagnosis Centre, Sacred Convent of the Annunciation,  
63071, Chalkidiki, Greece*

*kvavliak@ee.auth.gr, asymeon@iti.gr, g.karagiannis@artdiagnosis.gr, mitkas@eng.auth.gr*

## Abstract

*Semantic annotation and querying is currently applied on a number of versatile disciplines, providing the added-value of such an approach and, consequently the need for more elaborate – either case-specific or generic – tools. In this context, we have developed **Eikonomia**: an integrated semantically-aware tool for the description and retrieval of Byzantine Artwork Information. Following the needs of the ORMYLIA Art Diagnosis Center for adding semantics to their legacy data, an ontology describing Byzantine artwork based on CIDOC-CRM, along with the interfaces for synchronization to and from the existing RDBMS have been implemented. This ontology has been linked to a reasoning tool, while a dynamic interface for the automated creation of semantic queries in SPARQL was developed. Finally, all the appropriate interfaces were instantiated, in order to allow easy ontology manipulation, query results projection and restrictions creation.*

## 1. Introduction

Documenting and analyzing Byzantine Art aims to maintain and publish ones cultural heritage, thus proving the means for learning and comprehending ones history traversal path. The information needed for the complete documentation of artworks is usually multi-dimensional and comes from different sources and, for this reason, storing this information in Relational Database Systems (RDBMS) is extremely difficult and most of the times inefficient.

In the *ORMYLIA Art Diagnosis Center*, Byzantine artworks are analyzed with non-destructive methods such as infrared spectrums, x-rays and/or ultrasounds. The information derived from this kind of analysis, along with descriptive, interpretative, aesthetic and technical

information are currently stored in a well-designed Oracle Database System™. Nevertheless, latter reference to this data is extremely difficult, due to the inherent relationships between the data elements stored.

Towards this direction we have developed and present **Eikonomia**: an integrated semantically-aware tool for the description and retrieval of Byzantine Art information. The main aim of *Eikonomia* is to design and develop an ontology from the existing *Art-e-Fact* database (the legacy Oracle Database System™ developed in the Ormylia Center), adding this way semantic relations to the existing data and allowing for advanced data manipulation and querying. *Eikonomia* provides the user with the ability to easily manage the newly created ontology and synchronize it with the database system. The tool also employs a state-of-the-art reasoner (*Pellet*), in order to infer knowledge that has not been explicitly stated. Simple and complex querying functionality is supported through *Eikonomia*, while a practical interface for creating restrictions has also been developed.

This paper is structured as follows: Section 2 reviews background theory and related work; in Section 3 we present the functional facets of *Eikonomia*, while Section 4 demonstrates the developed system. Section 5 summarizes work done and concludes the paper with the authors' comments on future directions.

## 2. Background Information – Relevant Work

The World Wide Web Consortium [1] (W3C) is the main standardization body for the Web, issuing specifications, guidelines, software, and tools in an effort to transform the traditional Web into a Semantic Web (SW). Primary W3C recommendations include the Ontology Web Language [2] (OWL) and the SPARQL Query Language [3]. The cornerstone of the SW is the ontology, a structure that comprises classes and their

hierarchy, their properties and their instances, in order to describe data and their semantics. Ontologies support restrictions on data, relations among classes and reasoning tasks, such as classification and subsumption.

Numerous tools exist for the creation and inspection of ontologies, with Protégé [4] and TopBraid Composer [5] being the most widely accepted.

Based on ontologies, Description Logics [6] (DL) have developed a family of languages that represent knowledge in a structured (formal) manner, employing simple operands and existential or cardinality restrictions. Various tools exist employing DL primitives, aiming to extract knowledge that has not been explicitly stated within the ontology and to test ontology consistency. The most widely accepted ones, also known as *Reasoners*, are Pellet [7], Fact++ [8] and RacerPro [9]. Alternative knowledge extraction approaches also exist [10], while even data mining techniques have been applied to tackle the problem [11].

Advancing on ontology-related programming, the Jena [12] and Sesame [13] frameworks have been developed. These open-source APIs provide the programmer with the ability to easily store and load ontologies, perform semantic queries and/or reasoning tasks. In fact, to allow the transformation of data stored into relational databases to semantic aware information, tools such as D2RQ [14] and SquirrelRDF [15] have been developed.

On the other hand, preliminary work on the semantic representation of artworks already exists. The CIDOC Conceptual Reference Model [18] (CIDOC-CRM) is an official ISO standard that provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation. CIDOC-CRM is the culmination of over 10 years work and is intended to promote a shared understanding of cultural heritage information.

### 3. An overview of *Eikonomia* Facets

#### 3.1 CIDOC-CRM Extension Ontology

In order to ensure the unambiguous expression of terms in the *Art-e-Fact* database we have extended the CIDOC Conceptual Reference Model with terms capable of semantically annotating the database. This extension proved compulsory, since the *Art-e-Fact* database describes Byzantine artwork with highly technical details and spectrograms. The produced ontology by no means intends to replace CIDOC-CRM; nevertheless, it can be considered as an efficient effort to semantically annotate cultural data, since only a few implementations of CIDOC on real data are currently recorded. The CIDOC-CRM model has been enriched with a number of classes and properties, reaching the number of 257 classes and 298 properties (leaving out the inverse properties).

Due to the large number of classes and the complexity of CIDOC-CRM, as well as the complicated CIDOC onomatology, we refrain from presenting the whole ontology schema. In any case, one has to be familiar with the CIDOC architecture and the *Art-e-Fact* database to fully comprehend the mapping between the two schemas.

#### 3.2 Mapping the *Art-e-Fact* Database to Ontology

*Eikonomia* provides a direct transformation mechanism of the *Art-e-Fact* database to the extended ontology. Though the mapping mechanism is not as generic as other tools (D2RQ for example), it provides a simple way to map a database schema to an ontology, along with ‘write’ and ‘update’ capabilities to the derived ontology. The only prerequisite for this is knowledge of the database and ontology schemas. The mapping mechanism developed comprises three configuration files, which establish synchronization and ensure future compatibility, in case the ontology or database schema were altered. The first configuration file is responsible for the transformation of a new data tuple to ontology instances (and vice-versa), while the other two link the data with properties that existed in the same table row (second configuration file) and with respect to the table foreign keys (third configuration file).

**3.2.1. Creating Instances.** In order to transform all data residing in the database to ontology instances the first configuration file provides an one-to-one mapping function. Figure 1 illustrates the structure of such a configuration file.

Each line of the instance creation configuration file comprises four elements: the first refers to the table in the database, the second to the specific column of the table, while the next one refers to the class of the ontology that all values of the selected column should be mapped to. The last element of the configuration file refers to the table columns that hold the unique key(s) of the table, in order to ensure the unique URIs of the newly created instances.

```
ITEM PLACE E98_Storage_Place ITEM_ID;  
ITEM HIST_OWNER E39_Actor ITEM_ID;
```

**Figure 1.** Configuration file for instances creation.

The configuration file presented in Figure 1 will create instances equal to the number of rows of table *ITEM*. The URI of each instance will have the form of *ITEMPLACE98\_xxx\_cell\_value*, where *xxx* represents the unique key of each row, while *cell\_value* represents the value of the cell.

**3.2.2. Mapping instances of the same DB row.** Each file line comprises 7 elements. In a similar to the first configuration file manner, the first three elements specify

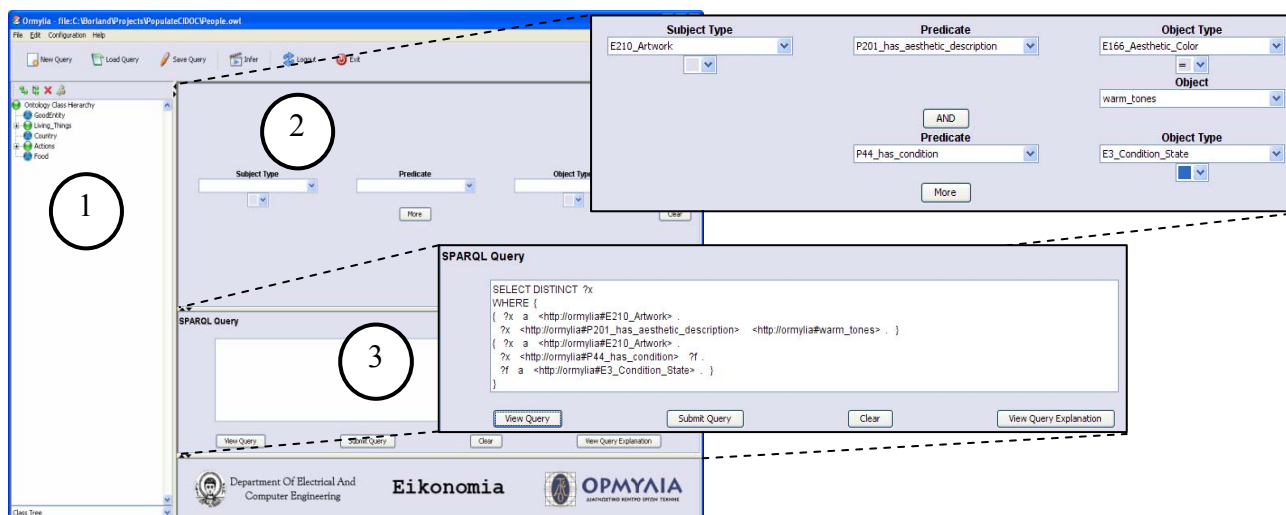


Figure 2. The *Eikonomia* main interface

the table, column and ontology class name for the first instance, while the fifth and sixth refer to the column and class name for the second instance. The fourth element specifies the object property between the two instances, while the last element point to the unique key(s) of the table.

**3.2.3. Mapping instances on their foreign keys.** The respective configuration file structure comprises 12 elements. The first three refer to the table, column and class name of the first instance respectively, while elements six, seven and eight refer to the second instance. The fourth and the ninth elements indicate the foreign keys of the two tables, while the fifth element defines the object property between the instances. Finally, elements following the ninth one indicate the primary keys of the two tables, separated with “\*\*”.

## 4. The developed system

### 4.1 Eikonomia main interface

*Eikonomia* provides a multi-functional user interface to facilitate users in their queries and experiments.

Figure 2.1 illustrates the main pane of *Eikonomia*, while 2.2 and 2.3 illustrate the ‘Semantic query creation’ pane and ‘Semantic query viewer and editor’ pane, respectively.

In the ‘Semantic query creation’ pane (Figure 2.2), the user can create semantic queries of the form *subject-predicate-object* with the assistance of drop-down menu fields. Each menu is created dynamically, with respect to the selections of the user, and takes into account the domain and range of the selected *predicate*. The ‘Semantic query creation’ pane also supports complex queries created through ‘AND’-‘OR’-‘NOT’ operands. The *subject* and *object* clauses of a query can either be a class or an instance of the ontology. In case an instance is selected, the equality (‘=’) and inequality (‘!=’) operands

are provided for the user to choose. Furthermore, when a datatype property is selected as the *predicate*, the available operands for the *object* are the following: ‘=’, ‘!=’, ‘>’, ‘>=’, ‘<’, ‘<=’, since in this case the *object* is a literal.

The created query is automatically transformed into the equivalent *SPARQL* query, which is projected in the ‘Semantic query viewer and editor’ pane (Figure 2.3), where the user is able to view and edit it directly in *SPARQL*. An explanation of the query in natural language is also provided. After reviewing the query, the user submits it to *Eikonomia*, and the results of the query are returned in an interactive manner; the user may select any instance and view all information about it, or edit/remove it. It should be denoted that the authors have no knowledge of a similar tool for creating semantic queries through a graphical user interface. *Eikonomia*, thus, provides an efficient way for semantic query creation to users that are not experts on *SPARQL*, or other SW technologies.

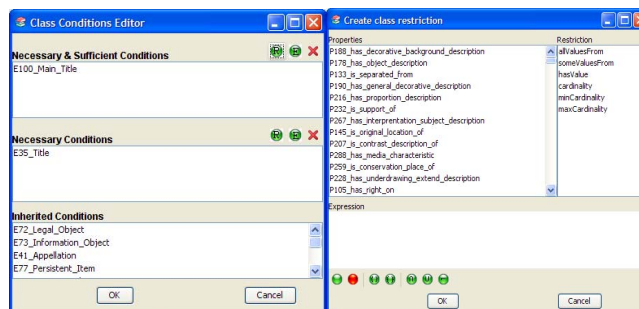


Figure 3. Restriction viewer & creator wizard

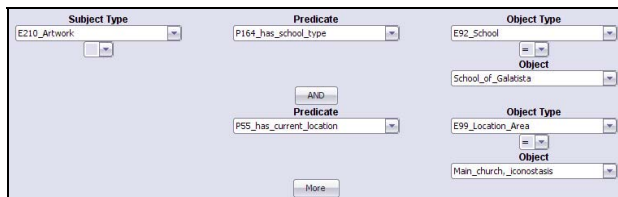
In a given *.owl* file one may define additional superclasses and equivalent classes and request reasoning against the restrictions that he/she has defined. *Eikonomia* provides an environment for viewing, creating and validating such restrictions (Figure 3).

## 4.1 Performing queries with Eikonomia

The automatic query creation mechanism of *Eikonomia* was exhaustively tested, in order to ensure its proper functionality. Both simple and complex queries were posted. Such a query is described below:

Let us assume that the user would like to find all artworks that follow the ‘*School of Galatista*’ methodology and are currently located in the ‘*Iconostasis of the main Galatista church*’.

The query is easily formulated by selecting *E\_210\_Artwork* as the *Subject*. Then, all properties related to the specific *Subject* are only provided in the *Predicate* drop-down menu for the user to select, which in turn specifies the available values for the *Object Type*. Since *E92\_School* is a class, the interface provides the user the ability to define equality/inequality to a specific instance value. Upon pressing the ‘*More*’ button, *Eikonomia* provides a second row of drop-down boxes for the user to create the ‘*AND*’ clause of the query. Figure 4 depicts the formulation of the query.



**Figure 4.** Formulating the desired query

## 4.2 Creating rules with Eikonomia

The added-value of the rule creation mechanism is also demonstrated through an example. Let us assume that *only* artist ‘*Ritzos Andreas*’ used ‘*gesso*’ in his artworks, i.e. an artwork made of gesso is surely created from him. For the user to define such a rule in the ontology he/she should create a new class named, say *Artworks\_Of\_Ritzos\_Andreas*, which must be a subclass of the existing *E210\_Artwork* class. Then, a necessary and sufficient condition for the newly created class should be created, in order to ensure that the Reasoner will successfully compute all instances that are instances of the class *E210\_Artwork* and consist of ‘*gesso*’, as instances of the new class. The Reasoner can now infer that all the relative instances belong to ‘*Ritzos Andreas*’, without any further actions required by the user. It should be denoted that, in order to perform such assumptions Reasoner should be enabled, otherwise the class *Artworks\_Of\_Ritzos\_Andreas* would be assigned no individuals.

## 5. Conclusions and Further Work

*Eikonomia* aspires to be the first applied effort towards introducing semantic technologies and their advantages to the Byzantine Art domain. Through

*Eikonomia*, the domain expert is expected to easily access and manage information on artworks and their interdependencies from a semantic perspective.

Future research efforts include the improvement of relation database to ontology mapping process, in order to ensure online synchronization. Additionally, with the assistance of domain experts, the ontology schema may be improved for the better representation of knowledge. Domain experts may also define the appropriate rules and restrictions, so that the Reasoner extracts advanced knowledge, but also preserves data consistency. Finally, the capabilities for ontology editing may be upgraded.

## 6. Acknowledgments

This work is supported by the 05AKMON83 project entitled "Development of a system for the management of knowledge resulting from the analysis and documentation of art objects" which is funded by the Greek General Secretariat of Research and Technology (GSRT).

## References

- [1] Semantic Web Coordination Group, <http://www.w3.org/2001/sw/CG/>.
- [2] F. v. Harmelen, S. Bechhofer, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider. "OWL Web Ontology Language Reference." W3C Recommendation. 10 February 2004.
- [3] E. Prud'hommeaux, W3C, A. Seaborne - Hewlett Packard Laboratories, W3C Candidate Recommendation, SPARQL Query Language for RDF, 6 April 2006.
- [4] Stanford University, Protégé – Ontology Editor and Knowledge acquisition tool, <http://protege.stanford.edu>
- [5] TopQuadrant Inc, TopBraid Composer™. Getting Started Guide Version 1.0. September 21, 2006.
- [6] F. Baader, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider. The Description Logic Handbook: Theory, implementation, and applications.
- [7] Mindswap Lab, University of Maryland, Pellet: An OWL DL Reasoner, <http://pellet.owldl.com/>.
- [8] FaCT++ OWL-DL Reasoner. <http://owl.man.ac.uk/factplusplus/>
- [9] Racor Systems GmbH & Co. KG, RacorPro User's Guide Version 1.9, 2005
- [10] R. Knappe, T. Andreasen and H. Bulskov, Department of Computer Science Roskilde University, Measuring Similarity From Conceptual Relations.
- [11] A. Bernstein, S. Hill, F. Provost, Center for Digital Economy Research, Stern School of Business, New York University Intelligent Assistance for the Data Mining Process: An Ontology-based Approach.
- [12] Jena – A Semantic Web Framework for Java. <http://jena.sourceforge.net/index.html>.
- [13] Sesame, <http://www.openrdf.org>
- [14] Freie University of Berlin, D2RQ V0.5 - Treating Non-RDF Databases as Virtual RDF Graphs. <http://sites.wiwi.fu-berlin.de/suhl/bizer/D2RQ/>.
- [15] SquirrelRDF, <http://jena.sourceforge.net/SquirrelRDF/>
- [16] International Council Of Museums, CIDOC Conceptual Reference Model. <http://cidoc.ics.forth.gr/>