Design and implementation of a semi-automatic tool for mapping source schemas to target ontologies

Ourania Smyrnaki

Thesis submitted in partial fulfillment of the requirements for the

Master of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Department of Computer Science
Voutes Campus, 70013 Heraklion, Crete
Greece

Thesis Supervisor: Prof. *Dimitris Plexousakis*

This work has been supported by the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH).

signatures

Design and implementation of a semi-automatic tool for mapping source schemas to target ontologies

Ourania Smyrnaki

Master Thesis

Computer Science Department, University of Crete

Abstract

As the Web evolves from the traditional Web (Web of documents) to the Web of Data (Semantic Web), there is a need for transforming data stored in XML schemas into semantically aware data. In addition, multiple metadata standards exist which vary in semantics and structure even if they belong to the same application domain. Since XML does not express semantics but rather the document structure, there is a lack of semantic interoperability between metadata standards. So, there is a need for expressing semantics, something not feasible with XML. Ontological approaches have been proposed in order to overcome the semantic heterogeneity and achieve metadata interoperability between heterogeneous sources. Ontologies have been created for the expression of semantics such as CIDOC-CRM , an ontology widely used in the cultural heritage domain as a common conceptual schema for information integration. In this thesis, a semi-automatic mapping tool for mapping source schemas to target ontologies is proposed. A mapping description language is used in order to describe the mappings between the source schema and the target ontology. This mapping tool simplifies the mapping process by suggesting target paths of the specified target ontology to the user and by re-using the "mapping-memories" stored in the mapping memory repository. In order to suggest mappings, the system performs schema matching between mapped and not mapped XML files in order to find crosswalks between those files. A mapped XML file is a file the elements of which have been already mapped to the elements in the target ontology and for which a mapping file exists in the mapping memory. Crosswalks found by the schema matching process are used by the mapping suggester, an essential component in the system that suggests mappings to the user. The proposed system also supports a friendly graphical user interface with additional components that analyze the source

and the target schema and is designed in order to be used by users without any programming knowledge.

Σχεδιασμός και υλοποίηση ενός ημι-αυτόματου συστήματος αντιστοίχησης πηγαίων σχημάτων (source schemas) σε οντολογίες στόχου (target ontologies)

Ουρανία Σμυρνάκη

Μεταπτυχιακή εργασία

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Περίληψη

Ενόψει της μετεξέλιξης του παραδοσιακού Ιστού από Ιστό εγγράφων σε Ιστό δεδομένων (Σημασιολογικό Ιστό – Semantic Web) , δημιουργείται η ανάγκη να μετατραπούν τα XML δεδομένα σε σημασιολογικά δεδομένα. Τα διάφορα σχήματα μεταδεδομένων, ακόμα και εάν ανήκουν στο ίδιο πεδίο εφαρμογής, διαφέρουν σημασιολογικά αλλά και ιεραρχικά, με αποτέλεσμα την έλλειψη σημασιολογικής διαλειτουργικότητας και την ύπαρξη ανάγκης για σημασιολογική συσχέτιση/διαλειτουργικότητα μεταξύ τους, κάτι που δεν είναι εφικτό με την XML. Για την αντιμετώπιση της σημασιολογικής ετερογένειας και την επίτευξη διαλειτουργικότητας μεταξύ ετερογενών πηγών έχουν προταθεί οντολογικές προσεγγίσεις. Έχουν δημιουργηθεί οντολογίες, όπως το CIDOC-CRM, που χρησιμοποιείται ευρέως στο χώρο της πολιτισμικής κληρονομιάς ως ένα κοινό εννοιολογικό μοντέλο για την ολοκλήρωση των πληροφοριών. Στην παρούσα εργασία, προτείνεται ένα ημι-αυτόματο εργαλείο για την αντιστοίχηση πηγαίων σχημάτων (source schemas) σε οντολογίες-στόχου (target ontologies). Γίνεται χρήση μίας περιγραφικής γλώσσας αντιστοίχησης για την περιγραφή των συσχετίσεων μεταξύ των στοιχείων του πηγαίου σχήματος και της οντολογίας στόχου. Το προτεινόμενο σύστημα επαναχρησιμοποιεί ήδη συσχετισμένα αρχεία δεδομένων (mapping files) με την βοήθεια ενός συστήματος συσχέτισης σχημάτων και προτείνει στον χρήστη υποψήφια μονοπάτια της οντολογίας στόχου, απλοποιώντας την διαδικασία αντιστοίχησης. Το σύστημα για να προτείνει τα υποψήφια μονοπάτια πραγματοποιεί συσχέτιση σχημάτων μεταξύ mapped XML αρχείων και not-mapped XML αρχείων. Για τα mapped XML αρχεία έχουν δημιουργηθεί αντιστοιχήσεις μεταξύ των στοιχείων τους και των στοιχείων της οντολογίας στόχου. Για τα not-mapped XML δεν έχουν δημιουργηθεί αντιστοιχήσεις ανάμεσα στα στοιχεία αυτών

και των στοιχείων της οντολογίας. Τα αποτελέσματα της παραπάνω συσχέτισης αξιοποιούνται από το ως άνω σύστημα το οποίο προτείνει στον χρήστη υποψήφια μονοπάτια της οντολογίας στόχου. Το σύστημα υποστηρίζει ένα φιλικό προς τον χρήστη γραφικό περιβάλλον με πρόσθετα εργαλεία (επιπλέον λειτουργίες) που αναλύουν το πηγαίο σχήμα και την οντολογία στόχου, και είναι ειδικά σχεδιασμένο για εύκολη και γρήγορη χρήση.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή μου τον κ. Δημήτρη Πλεξουσάκη για την αμέριστη συμπαράσταση, καθοδήγηση και βοήθειά του κατα τη διάρκεια των σπουδών μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών του Τμήματος Επιστήμης Υπολογιστών. Ιδιαιτέρως δε, τον ευχαριστώ που δέχθηκε να είναι επιβλέπων Καθηγητής μου στην μεταπτυχιακή μου εργασία, για τη συνεχή και σημαντική αρωγή του, το έμπρακτο ενδιαφέρον για την πορεία της έρευνάς μου και την άμεση και αποτελεσματική διευθέτηση και επίλυση κάθε θέματος σχετικού με την εκπόνηση της εργασίας μου.

Ιδιαιτέρως ευχαριστώ τον κ. Μάρτιν Ντερρ, ερευνητή στο ΙΤΕ, μέλος της τριμελούς επιτροπής της μεταπτυχιακής μου εργασίας, με τον οποίο συνεργαστήκαμε κατά τη διάρκεια της έρευνας και συγγραφής. Τον ευχαριστώ για την πολύτιμη βοήθειά του σε κάθε στάδιο της έρευνάς μου, την παροχή χρήσιμων και καθοριστικής σημασίας οδηγιών, συστάσεων και προτάσεων, και για την άμεση και ουσιαστική ανταπόκρισή του σε όποιο ζήτημα υπήρχε.

Ευχαριστώ επίσης τον Καθηγητή Τζίτζικα Ιωάννη για την θετική ανταπόκρισή του και προθυμία του να είναι μέλος της τριμελούς επιτροπής, για τη διάθεσή του να παρέχει τη βοήθειά του και τις γνώσεις του για τη διαμόρφωση της μεταπτυχιακής μου εργασίας.

Παράλληλα, θα ήθελα να ευχαριστήσω το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την παροχή υλικοτεχνικής υποδομής κατά την εκπόνηση της εργασίας μου και τα μέλη του Εργαστηρίου Πληροφοριακών Συστημάτων για τη στήριξή τους.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξη της κατά τη διάρκεια των σπουδών μου καθώς και το φιλικό μου περιβάλλον.

Contents

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The World Wide Web which started as a web of documents is evolving to a web of data. The web was initially based on web pages - documents linked together by use of hyperlinks, allowing simple keyword searching as a result of low level structures. Search engines "*fail to provide the epistemological and historical context of a question which gives results a meaning; they are designed as a tool for information aggregation not knowledge integration*"[1]. Following the highly increasing needs to find relations between data, itself, and not documents, the Web of Data emerged.

The web of data, which includes the integration of data online, is enabled by technologies such as RDF, where relations between data in XML or RDB form, are expressed as triplets (subject-property-object). However, the problem of heterogeneity of data representations hinders full realization of web of data. In order to achieve the full potential of linked data, there is a need to achieve metadata interoperability by the implementation of harmonized standards. What we are trying to do, is to interconnect data and at the same time to overcome the issues created by the metadata standards. Even in the same sectors, there are various different metadata standards, that is, different ways to express data.

There is a tendency to integrate all data coming from heterogeneous sources in order to create a global knowledge network. The methodology which is followed, is the transformation of the data to RDF format by use of ontologies. Mapping the data of XML and RDB formats, to RDF ones. In order to achieve that, we must first map our data to a target ontology. This mapping is not an easy procedure, especially if made manually. To propose a solution to the above, we have built and implemented a semi-automatic mapping tool, which accepts an XML document file as input and a target schema ontology and performs the mapping between the source and the target

[1] Doerr, M., Iorizzo, D., "The Dream of a Global Knowledge Network – A New Approach" , Journal on Computing and Cultural Heritage, vol.1, Iss 1, pp. 1-23. ACM, 2008.

schema. Our tool supports a friendly Graphical User Interface that consists of a source analyzer, a target schema analyzer with a target schema visualizer, a target schema path editor with a target schema validator. It can be used by non IT experts that may not have programming education/knowledge. It produces automatically mapping files between the source schema and the target schema ontology and stores them in the mapping memory repository. It follows a reuse strategy by re-using the mappings stored in the mapping memory "mapping memories" with the assistance of a schema matching tool. It suggests mappings to the user by exploiting the mappings stored in the mapping memory.

The mapping memory is a repository that contains mapping files of already mapped XML document files. A mapping file is written in a specific mapping description language and describes the mappings between the elements in the source schema and elements in the target schema. A mapped XML file is a document file the elements of which have already been mapped to the elements in the target schema ontology. The system performs a schema matching between a not mapped XML file and a mapped XML file. Correspondences/Crosswalks found by the schema matching are used by the mapping suggester (an essential component described later in the thesis) in order to suggest mappings to the user. User feedback is taken into consideration during the mapping process.

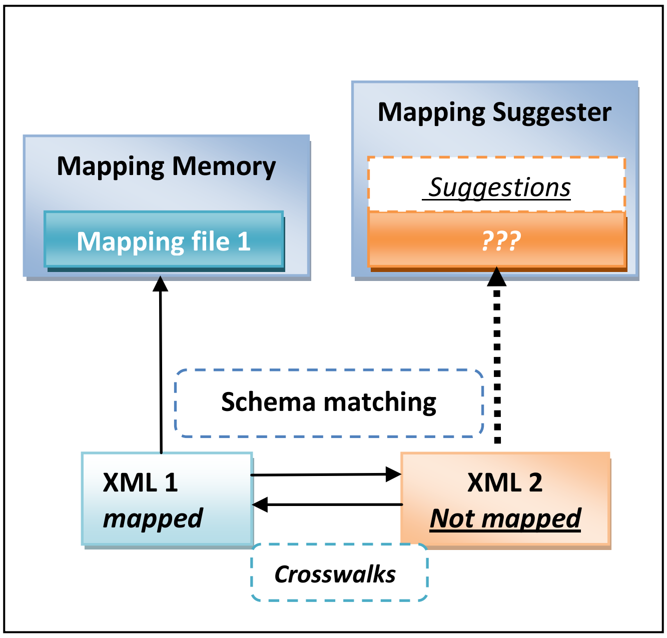


Figure 1.1 System architecture

Also, our system proposes a Learning method in order to improve the accuracy of the schema matching process and reduce the false negatives matches.

Further information is included in Chapter 3.

The thesis consists of the following parts:

In the second chapter, reference is made to the existing related work and background knowledge. There are four levels of heterogeneity in the process of data integration. Some of the different metadata standards and solutions for semantic interoperability and data integration are mentioned, the way mapping is performed is described and some of the existing mapping tools are presented.

In the third chapter, our system is analyzed in detail.

In chapter three, an evaluation of the system is presented. We use evaluation criteria such as Precision, Recall and F-measure in order to test our results.

A reference to Future Work and the Conclusion of the thesis follow.

Chapter 2

2 Background Knowledge and Related Work

2.1 Types of heterogeneity

Data from multiple sources are characterized by multiple types of heterogeneity, which are [2,3,4,5,6].

- Syntactic heterogeneity, formats of data differ.
- Schematic or structural heterogeneity, the structures to store data differ in data sources. Schematic heterogeneity happens mostly in structured databases.
- Semantic Heterogeneity. Metadata schemes express the meaning of data in different ways resulted in.
- System Heterogeneity reflects differences in operating systems, hardware and platforms.

2.2 Semantic Interoperability

There are many metadata standards which are structured in different ways resulting in heterogeneity. It has been suggested to adopt one single standard. In addition, solutions to achieve metadata standards' interoperability include the use of metadata derivation, application profiles, metadata-crosswalks (metadata matching), metadata

[2] http://en.wikipedia.org/wiki/Ontology-based_data_integration

[3] Koutrika, G., "Heterogeneity in Digital Libraries: two sides of the same coin", Delos Network of Excellence on digital libraries, Iss 3. DELOS Network of Excellence Newsletter, 2005. Available from: http://www.delos.info/files/pdf/newsletter/delos-newsletter-issue3.pdf

[4] David, G., "Understanding Structural and Semantic Heterogeneity in the Context of Database Schema Integration", In: Proceedings of the 6th Conference in the Dept. of Computing (Journal of the Dept. of Computing), Iss. 4, pp. 29-44. 2005.

[5] Sheth, A. P., " Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics, in Interoperating Geographic Information Systems ", Interoperating Geographic Information Systems, vol. 495, pp. 5-29. Springer, 1999.

[6] Alemu, G., Stevens, B., Ross, P., "Towards a conceptual framework for user-driven semantic metadata interoperability in digital libraries: A social constructivist approach.", New Library World, vol. 113, Iss. 1/2, pp. 38-54. Emerald, 2012.

registries, the use of semantic web technologies. However, even the above methods do not offer the required level of interoperability in relation to cross-searching, content sharing and information integration. [7]

CIDOC CRM (Conceptual Reference Model) is an “ontological approach to semantic interoperability of metadata”.[8] So, instead of using a common schema for data capturing, CIDOC CRM is a “semantic approach to integrated access”.

2.3 Mapping Technology

In this section an XML file is mapped to the CIDOC-CRM ontology. The source schema is an XML document file that describes a painting “La Primavera/The Spring” and the target schema is CIDOC-CRM .The XML document file is encoded in LIDO[9], a metadata standard used in the museum domain. The XML file is available at [10]:

Figure 2.1 :”La Primavera” painting available online at: *http://en.wikipedia.org/wiki/File:Botticelli-primavera.jpg*

[7] Alemu, G., Stevens, B., Ross, P., “Towards a conceptual framework for user-driven semantic metadata interoperability in digital libraries: A social constructivist approach.”, New Library World, vol. 113, Iss. 1/2, pp. 38-54. Emerald, 2012.

[8] Doerr, M.,”The CIDOC CRM, an Ontological Approach to Semantic Interoperability of Metadata”, AI Magazine, vol. 24, pp. 75-92, 2003.

[9] LIDO Specification V1.0 , November 2010, Available online at: http://www.lido-schema.org/schema/v1.0/lido-v1.0-specification.pdf

[10] LIDO XML example “La Primavera (painting), Available online at: http://www.lido-schema.org/documents/examples/LIDO-Example_FMobj00154983-LaPrimavera.xml

Text Information about the Primavera Painting:
[11]“*Primavera* <…> is a tempera panel painting by Italian Renaissance artist Sandro Botticelli” [12]Sandro Botticelli was born in 1445 and he is an Italian painter.

Table 2.1 Text information (La Primavera painting)

The above information is written in a text form but since we would like the information to be machine readable, we encode it in LIDO metadata standard based on the example described in [13].

XML is both human-readable and machine-readable.

The table below describes LIDO XML document file:

Data encoded in LIDO metadata standard (Source schema) [xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<lido:lido>
  <lido:lidoRecID lido:type=""></lido:lidoRecID>
  <lido:descriptiveMetadata xml:lang="en">
      <lido:objectClassificationWrap>
        <lido:objectWorkTypeWrap>
          <lido:objectWorkType>
            <lido:term>painting</lido:term>
          </lido:objectWorkType>
        </lido:objectWorkTypeWrap>
        <lido:classificationWrap>
          <lido:classification>
            <lido:term>panel painting</lido:term>
          </lido:classification>
        </lido:classificationWrap>
      </lido:objectClassificationWrap>
      <lido:objectIdentificationWrap>
        <lido:titleWrap>
          <lido:titleSet>
            <lido:appellationValue lido:pref="preferred">La Primavera /
Spring</lido:appellationValue>
          </lido:titleSet>
```

[11] http://en.wikipedia.org/wiki/Primavera_(Painting)

[12] http://en.wikipedia.org/wiki/Sandro_Botticelli

[13] http://www.lido-schema.org/documents/examples/LIDO-Example_FMobj00154983-LaPrimavera.xml

```
        </lido:titleWrap>
      </lido:objectIdentificationWrap>
      <lido:eventWrap>
        <lido:eventSet>
          <lido:event>
            <lido:eventType>
              <lido:term>creation</lido:term>
            </lido:eventType>
            <lido:eventActor>
              <lido:actorInRole>
                <lido:actor lido:type="person">
                  <lido:nameActorSet>
                    <lido:appellationValue lido:pref="preferred">Botticelli,
Sandro</lido:appellationValue>
                  </lido:nameActorSet>
                  <lido:nationalityActor>
                    <lido:term>Italien</lido:term>
                  </lido:nationalityActor>
                  <lido:vitalDatesActor>
                                                            <lido:earliestDate
lido:type="estimatedDate">1445</lido:earliestDate>
                  </lido:vitalDatesActor>
                  <lido:genderActor>male</lido:genderActor>
                </lido:actor>
                <lido:roleActor>
                  <lido:term>painter</lido:term>
                </lido:roleActor>
              </lido:actorInRole>
            </lido:eventActor>
            <lido:eventMaterialsTech>
              <lido:materialsTech>
                <lido:termMaterialsTech lido:type="material">
                  <lido:term>tempera</lido:term>
                  <lido:term lido:addedSearchTerm="yes">color
material</lido:term>
                </lido:termMaterialsTech>
              </lido:materialsTech>
            </lido:eventMaterialsTech>
          </lido:event>
        </lido:eventSet>
      </lido:eventWrap>
    </lido:descriptiveMetadata>
    <lido:administrativeMetadata xml:lang="en">
      <lido:recordWrap>
        <lido:recordID lido:type=""></lido:recordID>
        <lido:recordType></lido:recordType>
        <lido:recordSource></lido:recordSource>
      </lido:recordWrap>
    </lido:administrativeMetadata>
</lido:lido>
```

Table 2.2 LIDO XML example (sample) available online at *http://www.lido-schema.org/documents/examples/LIDO-Example_FMobj00154983-LaPrimavera.xml*

If we would like to map the LIDO XML file to CIDOC-CRM, the mapping would be as follows:

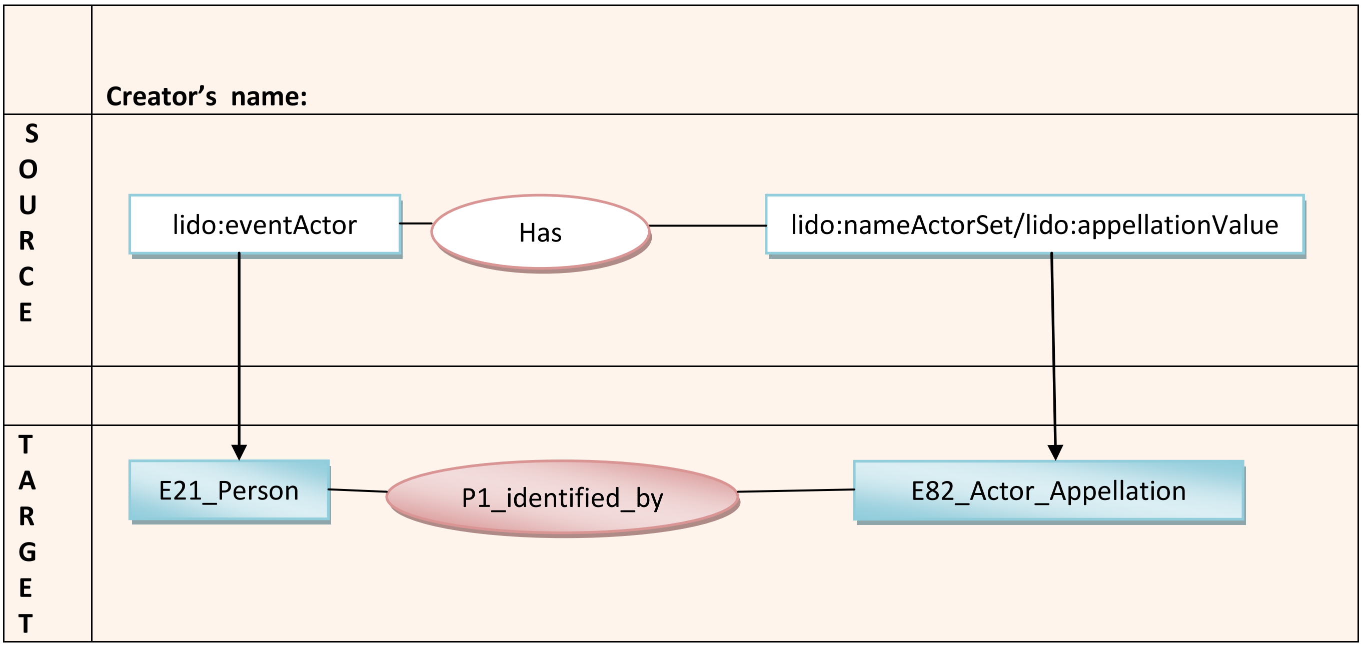


Figure 2.2 Example 1: Mapping of LIDO to CIDOC-CRM

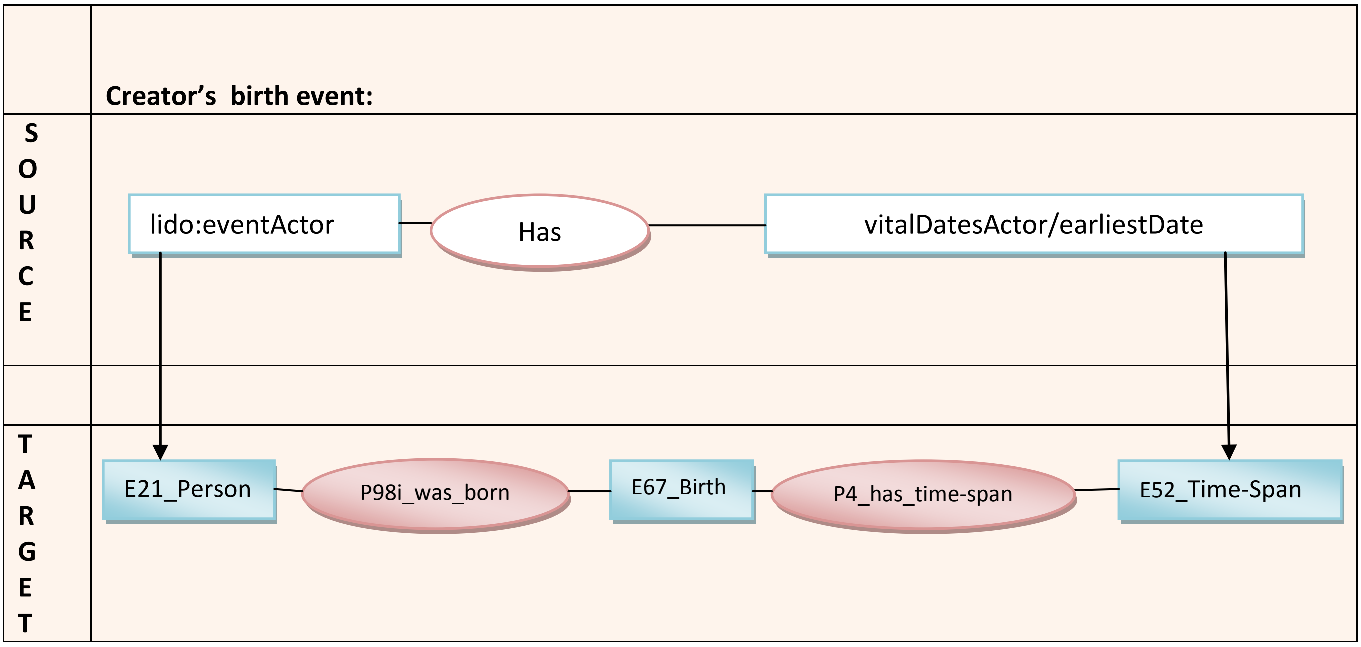


Figure 2.3 Example 2: Mapping of LIDO to CIDOC-CRM

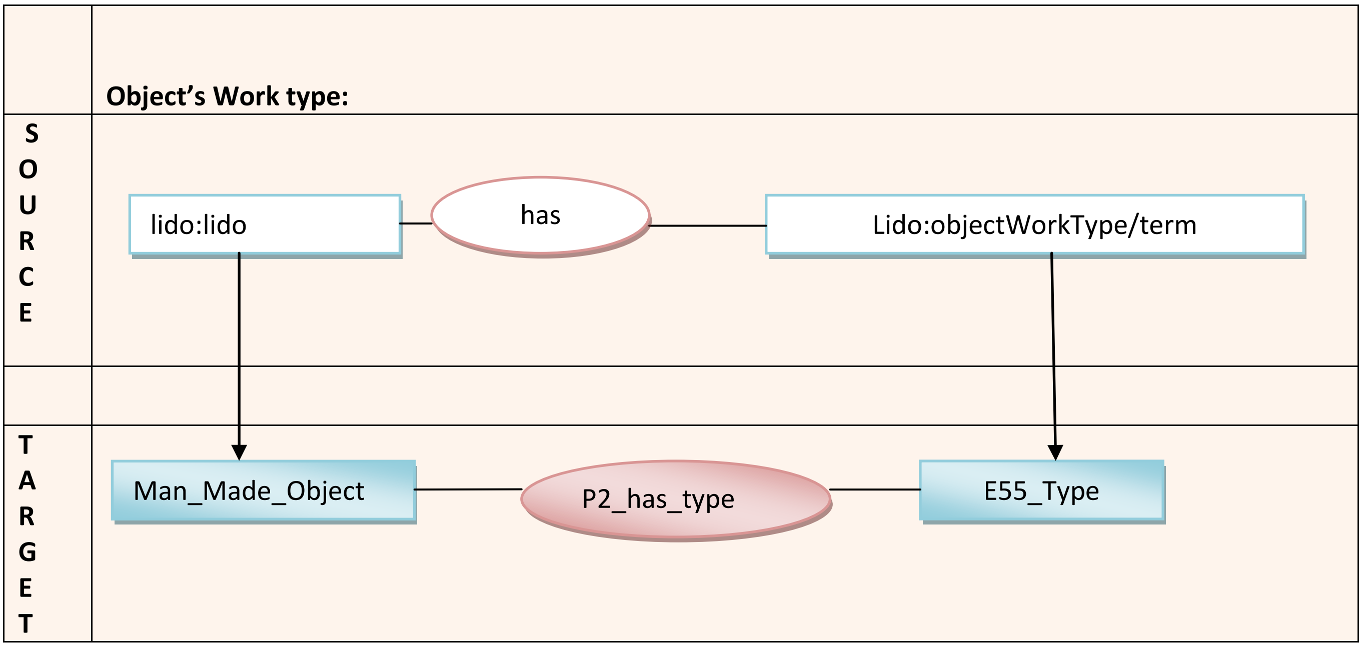


Figure 2.4 Example 3: Mapping of LIDO to CIDOC-CRM

In order to describe the above mapping, we use a mapping description language that was proposed here:[14]

2.4 Mappings of CIDOC-CRM

The past years a lot of effort has been done in describing the semantic mappings of various metadata standards (such as Dublin Core[15,16], EAD[17,18,19,20], VRA Core 4.0 [21,]

[14] Kondylakis, H., Doerr, M., Plexousakis, D., "Mapping Language for Information Integration", Technical Report 385, ICS-FORTH, 2006.

[15] Doerr M., "Mapping of the Dublin Core Metadata Element Set to the CIDOC CRM", Technical Report 274, ICS-FORTH, 2000. Available from: *http://www.cidoc-crm.org/docs/dc_to_crm_mapping.pdf*

[16] Kakali, C., Lourdi, I., Stasinopoulou, T., Bountouri, L., Papatheodorou, C., Doerr, M., Gergatsoulis, M., "Integrating Dublin Core metadata for cultural heritage collections using ontologies", In: Proceedings of the International Conference on Dublin Core and Metadata Applications, pp. 128-139, Singapore, 2007.

[17] Theodoridou, M.,Doerr, M., "Mapping of the encoded Archival Description DTD Element Set to the CIDOC-CRM", Technical Report 289, ICS-FORTH, 2001, Available from: http://ics.forth.gr/isl/publications/paperlink/ead.pdf

[18] Bountouri, L., Gergatsoulis, M., Papatheodorou, C., "Mapping EAD to CIDOC CRM", In: Proceedings of the 21st SIG meeting and 15th FRBR - CIDOC CRM Harmonization meeting Workshop on Conceptual Modelling for Archives, Libraries and Museums, Finland, 2010.

[22], MPEG-7 [23], DCCAP[24], TEI[25], LIDO[26], Midas[27]) to the CIDOC CRM. In addition, mappings from other formats to the CIDOC – CRM have been proposed that contain mappings to the semantic model CIDOC CRM such as texts and other digital formats[28], data structures[29], data examples (e.g. Epitaphios GE34604[30]), data dictionaries (e.g. AMICO[31]), bibliographic formats (e.g. UNIMARC[32]) and MDA data Spectrum[33] .

[19] Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou, C., Doerr, M., Gergatsoulis, M., "Ontology-based Metadata Integration in the Cultural Heritage Domain", Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, Lecture Notes in Computer Science,vol. 4822, pp. 165-175. Springer, 2007.

[20] Bountouri, L., Gergatsoulis, M., "Mapping Encoded Archival Description to CIDOC CRM." In: Proceedings of the 1st Workshop on Digital Information Management, Corfu, Greece, pp. 8-25. Ionian University, 2011.

[21] Gaitanou, P., Gergatsoulis, M., "Mapping VRA Core 4.0 to the CIDOC/CRM ontology", In: Proceedings of the 1st Workshop on Digital Information Management, Corfu, Greece, pp. 26-38. Ionian University, 2011.

[22] Gaitanou, P., Gergatsoulis, M., "A Semantic Mapping of VRA Core 4.0 to the CIDOC Conceptual Reference Model", In: Proceedings of Metadata and Semantic Research 5th International Conference, MTSR 2011, Izmir, Turkey, Communications in Computer and Information Science, vol. 240, pp. 387-399. Springer, 2011.

[23] Angelopoulou, A., Tsinaraki, C., Christodoulakis, S., "Mapping MPEG-7 to CIDOC/CRM", Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, vol. 6966, pp. 40-51. Springer, 2011.

[24] Lourdi, I., Papatheodorou, C., Doerr, M., "Semantic Integration of Collection Description Combining CIDOC/CRM and Dublin Core Collections Application Profile", D-Lib Magazine, vol. 15, number 7/8, 2009.

[25] Eide, Ø., Ore, C-E.," Mapping of TEI to CIDOC-CRM", 2007. Available from: http://www.edd.uio.no/artiklar/tekstkoding/tei_crm_mapping.html

[26] Koutraki , M., Doerr , M., "Mapping LIDO v0.7 to CIDOC-CRM v5.0.1", Working paper, FORTH-ICS, 2010. Available from: http://www.cidoc-crm.org/docs/mappings/Mapping_lido_v2.doc

[27] MIDAS to CRM, Available from: http://www.cidoc-crm.org/docs/midas_map.xls

[28] Généreux,M., Niccolucci, F.,"Extraction and mapping of CIDOC-CRM encodings from texts and other digital formats", In: Proceedings of the 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST, 2006.

[29] Doerr, M., "Mapping a Data Structure to the CIDOC Conceptual Reference Model", ICS-FORTH, Heraklion, Crete, 2002. Available from: http://www.cidoc-crm.org/docs/mapping.ppt

[30] Doerr, M., Dionissiadou, I.,"Data Example of the CIDOC Reference Model - Epitaphios GE34604", 2007. Available from: http://www.cidoc-crm.org/docs/epitafios1.htm

[31] Doerr, M., "Mapping of the AMICO data dictionary to the CIDOC CRM", Technical Report FORTH-ICS/TR-288, 2000. Available from: http://www.cidoc-crm.org/docs/mappingamicotocrm.rtf

[32] Boeuf, P.L., "Mapping from UNIMARC Bibliographic to CIDOC CRM", Version 1.0, 2006. Available from: http://www.cidoc-crm.org/docs/UNIMARCB_CRM.zip

[33] "MDA Spectrum CIDOC CRM mapping", 2003, Available from: http://www.cidoc-crm.org/docs/MDA%20Spectrum_CIDOC_CRM_mapping.pdf

2.5 Mapping tools

In this section we describe some of the mapping tools that map a source schema (RDB or XML) to a target ontology.

2.5.1 AMA TOOL

Archeological and cultural heritage data are not organized and linked together. Rather, they are digitally stored and archived in large amounts in different institutions[34]. The variety of documentation standards or the complete absence of any standards, constitute factors which impair usability of the data and discourage any effort to connect all these datasets. In addition, the cultural professionals and computer scientists have not found common channels of communications regarding cultural concepts[35].

AMA project run by EPOCH, attempts to overcome the above problems by use of the CIDOC-CRM. CIDOC-CRM works on information integration in a form understandable by all experts of the field, representing a broad consensus on ontological commitment, a thing that made CIDOC CRM an ISO standard for cultural information[36]. AMA project chose CIDOC CRM because it can describe "implicit and explicit concepts and relationships used in cultural heritage documentation"[37] aiming at achieving a shared understanding of cultural heritage information[38]. As a matter of fact, when the project was initiated, they wanted to promote the use of CIDOC CRM as a reference model by heritage professionals.

The above mentioned project developed an open source tool to achieve data interchange and interoperability of different information repositories. The AMA project tool aims to facilitate the mapping of different data models to a common

[34] "European Network of Excellence in Open Cultural Heritage (EPOCH)", Available from: http://www.epoch-net.org/index.php?option=com_content&task=view&id=222&Itemid=338. Accessed 10 December 2013.

[35] Doerr, M.,"The CIDOC CRM, an Ontological Approach to Semantic Interoperability of Metadata", AI Magazine, vol. 24, pp. 75-92, 2003.

[36] Doerr, M.,"The CIDOC CRM, an Ontological Approach to Semantic Interoperability of Metadata", AI Magazine, vol. 24, pp. 75-92, 2003.

[37] The CIDOC Conceptual Reference Model, available online at http://www.cidoc-crm.org/

[38] The CIDOC Conceptual Reference Model, available online at http://www.cidoc-crm.org/

standard using the CIDOC CRM ontology[39]. Datasets pertaining to structured data as excavation data, archaeological, cultural data, museum collections, or even non-structured data, like free text data, are mapped to a CIDOC CRM compatible form providing at the same time the information entailed to convert the individual datasets to the CIDOC-CRM compliant structure[40,41].

It's worth mentioning that the AMA project includes the development of the AMA mapping tool, mentioned above, the APA text tool, which marks up unstructured text documents recursively, and a semantic web database which manages (stores, queries, returns) semantic information.

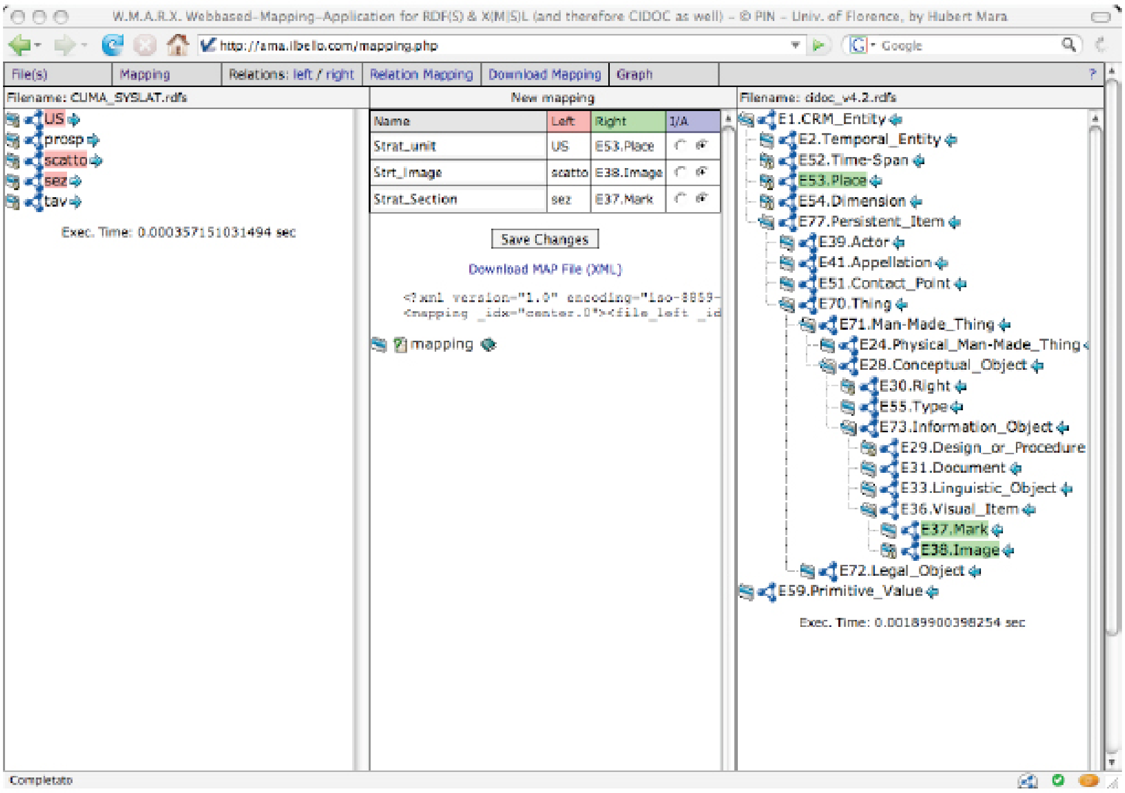
Regarding the AMA mapping tool, the mapping is performed according to the following steps:

1. Uploading of a source and a target schema in any XML format (RDFS is suggested for the target schema)
2. Mapping pairs of classes (1:1),(1:n),(n:m)
3. Defining the relations between the classes in the target schema using the properties of the mapped class

[39] Felicetti, A., Hubert, M., "Semantic Maps and Digital Islands: Semantic Web technologies for the future of Cultural Heritage Digital Libraries", In: Proceedings of the 5th European Semantic Web Conference , pp. 51-62, Tenerife, Spain, 2008.

[40] "AMA – Archive Mapper for Archaeology", EPOCH European Network of Excellence in Open Cultural Heritage, Available from: http://www.epoch-net.org/index.php?option=com_content&task=view&id=222&Itemid=338,
Accessed 10 December 2013.

[41] Felicetti, A., Hubert, M., "Semantic Maps and Digital Islands: Semantic Web technologies for the future of Cultural Heritage Digital Libraries", In: Proceedings of the 5th European Semantic Web Conference , pp. 51-62, Tenerife, Spain, 2008.



Exec. Time: 0.000357151031494 sec

Name	Left	Right	I/A
Strat_unit	US	E53.Place	○ ●
Strt_Image	scatto	E38.Image	○ ●
Strat_Section	sez	E37.Mark	○ ●

```
<?xml version="1.0" encoding="iso-8859-
<mapping _idx="center.0"><file_left _id
```

Exec. Time: 0.00189900398254 sec

Figure 2.5 : Graphical User Interface of AMA tool *(figure available at* *http://image.ntua.gr/swamm2006/SIEDLproceedings.pdf pg. 55)*

The interface of the AMA Mapping tool is written in PHP language where the mapping template is defined and can be exported to be used[42]. It gives the potential to upload XML starting schema, perform mapping, to upload mapping ontologies other than CIDOC CRM, and advanced features such as creation of new entities to indicate relations, creations of shortcuts, graphic visualization of relations produced during the mapping process[43].

[42] Eide, Ø., Felicetti, A., Ore, C.E., Andrea, A.D., Holmen, J., "Encoding Cultural Heritage Information for the Semantic Web". In: Proceedings of the EPOCH Conference on Open Digital Cultural Heritage Systems, pp. 1-7,2008.

[43] Eide, Ø., Felicetti, A., Ore, C.E., Andrea, A.D., Holmen, J., "Encoding Cultural Heritage Information for the Semantic Web". In: Proceedings of the EPOCH Conference on Open Digital Cultural Heritage Systems, pp. 1-7,2008.

2.5.2 ANNOCULTOR

AnnoCultor is a technical tool which converts databases and XML files to RDF and semantically tags (enriches) them with links to various vocabularies[44].

It is built in Java code, was developed by the E-Culture project and was released as open source under GPL license. The Annocultor tool offers conversion infrastructure and implements several conversion rules[45].

At the University of Amsterdam, they were trying to develop a semantic web application/search engine (ECULTURE) for searching the collection of many and various cultural institutions. During the process of building the application, they had to deal with the technical conversion problems. That is, they needed to convert the source data to the target schema and make alignments to standard vocabularies and terms[46]. It has been used for the conversion of collection in Louvre, Rijksmuseum Amsterdam etc.

An example of how the Annocultor works is described below[47]:

If we have the following values

dcterms:spatial = Venice

dc:date = 15e siècle

AnnoCultor interpretes them, and searches in vocabularies (that is, specialized databases of places and periods) to generate the corresponding terms (e.g. in Geonames and in Annocultor Time Ontology) and adds the relevant links to the terms.

- link to place: http://sws.geonames.org/3164603/
- link to period http://annocultor.eu/time/14xx

In addition, it extracts more info about the terms from the databases (vocabularies entries) and adds the following links (semantic tags).

place labels: βενετία, velence, венеция, venice, etc

geo coordinates: 45.43861, 12.32667

[44] Walkowska, J., Werla, M., “Advanced Automatic Mapping from Flat or Hierarchical Metadata Schemas to a Semantic Web Ontology”, Theory and Practice of Digital Libraries, Lecture Notes in Computer Science, vol. 7489, pp 260-272. Springer, 2012.

[45] Omelayenko, B., “Porting Cultural Repositories to the Semantic Web”, In: Proceedings of the First Workshop on Semantic Interoperability in the European Digital Library , pp. 14-25, Tenerife, Spain, 2008.

[46] Ibid.

[47] Omelayenko, B., ”Semantic Tagging at large scale”, available from http://borys.name/blog/semantic_tagging_of_europeana_data.html, Accessed 19 November 2013.

time labels: 15th, 15de eeuw, 15e siècle, 15й век, etc

temporal endpoints: 1401-01-01T00:00:00Z, 1500-12-30T23:59:59Z

AnnoCultor[48,49,50,51]	
Pros	Cons
Provides conversion rules to convert - XML into target RDF resources, - XML tags into RDF properties, - XML tag values into RDF resources and literals.	It is not a user-friendly tool running only from the command line
Provides mechanisms for semantic data enrichment - tagging: - It loads vocabulary of terms (specialized databases), - It finds XML tag values in these vocabularies, - And maps the findings to the vocabulary terms.	It is not an automatic converter and you need to know exactly how your XML document should be represented in RDF, you need to write a separate rule for each XML tag that will be then converted to one or more RDF triples;
Simplifies data analysis and quality assurance by showing all XML paths, their most used values and their uniqueness and by generating conversion reports with statistics, XML tags etc. missed in the conversion or not found in the databases etc.	Mapping entails manual provision of term labels and java code.
It can run on Windows, Unix, Mac.	There is a need to provide vocabularies

[48] "Annocultor", EPOCH European Network of Excellence in Open Cultural Heritage. Available from: http://semium.org/overview.html, Accessed 10 December 2013.

[49] Geser, G., "STERNA (Semantic Web- Based Thematic European Reference Network Application)", Technology Watch Report, January 2009, Available from: http://www.salzburgresearch.at/wp-content/uploads/2010/10/sterna_tech_watch_report_layoutiert_web.pdf

[50] Leroi, M. V., Holland, J., "Access to cultural Heritage networks across Europe", ATHENA WP4 Technical Meeting 2010. available online at http://151.12.58.141/athena_mw14/index.php?en/111/events/110/paris-athena-wp4-technical-meeting

[51] https://github.com/europeana/tools/blob/master/annocultor_solr4/src/site/apt/overview.apt,Accessed in November 2013

	and the tagging is not done automatically.

Table 2.3 Graphical User Interface of AMA tool

2.5.3 RDFer[52] (by British Museum)

RDFer is the in-house application implemented by the British Museum. In fact, it is the British Museum’s CRM mapping tool. It transforms XML files from the collection system of the Museum to RDF triples. It uses XPath syntax. It entails programming knowledge to use and understand the tool.

Below you may see the xml configuration file defining the mapping between the Museum's data model and the CRM.

```
<mapping match="{//bm_object[bm_object_part/_[mus_obj_parts='1' and
bm_alias_admin_no/_/bm_admin_type='WEB']]}"

namedgraph="&id;object/{bm_prn}/graph">
 <resource>
 <identifier prefix="http://collection.britishmuseum.org/id/object/"
value="{bm_prn}"/>

 <type value="http://erlangen-crm.org/current/E22_Man-Made_Object"></type>
 <!-- IDENTITIFERS -->
 <!-- Merlin PRN -->
 <triple object="{bm_prn}/prn" predicate="crm:P48_has_preferred_identifier"
prefix="http://collection.britishmuseum.org/id/object/"></triple>
 <resource>
```

[52] Oldman, D., Mahmud, J., Alexiev, V., “ The Conceptual Reference Model Revealed, Quality contextual data for research and engagement: A British Museum case study”, 2013 . available online at https://confluence.ontotext.com/download/attachments/33325240/mapping+manual+for+endpoint+site+draft+0.98a.pdf?version=1&modificationDate=1386147054000/

<identifier prefix="http://collection.britishmuseum.org/id/object/"
value="{bm_prn}/prn"></identifier>
<type value="http://erlangen-crm.org/current/E42_Identifier"></type>
<triple object="http://collection.britishmuseum.org/id/thesauri/identifier/prn"
predicate="crm:P2_has_type"></triple>
<triple predicate="rdfs:label" value="{bm_prn}"></triple>
</resource>

Table 2.4 Configuration file of RDFer tool

2.5.4 XML to RDF Transformation Tool

CIDOC CRM/ FRBR XML representations are converted to RDF representation based on the DTD and RDFS files of CIDOC CRM version 5.0.2 and FRBR version 1.0.1[53]. The mapping of the XML files to the target schema (CIDOC CRM etc) is also based on a mapping description language[54]. The tool is a java application and you may see a screenshot of the system below:

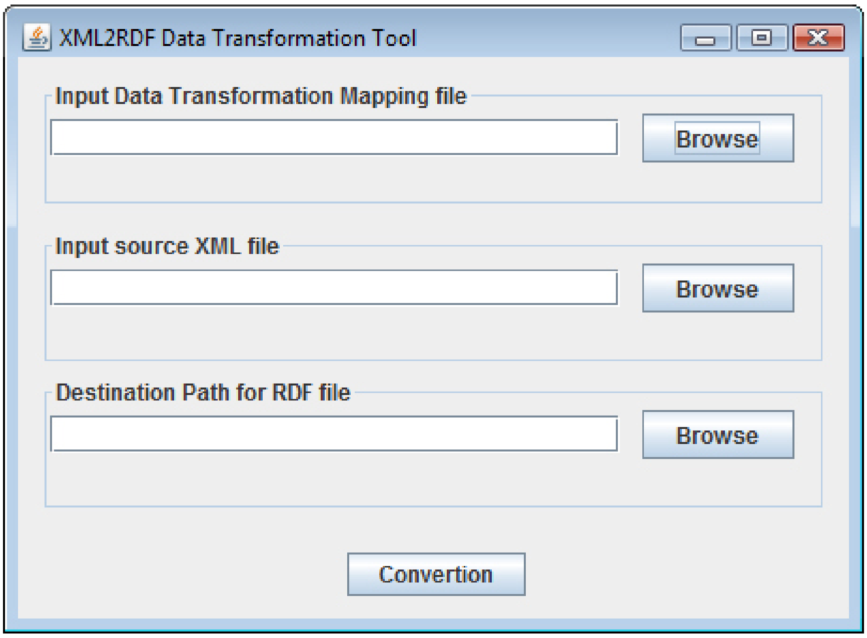

XML2RDF Data Transformation Tool
Input Data Transformation Mapping file
Browse
Input source XML file
Browse
Destination Path for RDF file
Browse
Convertion

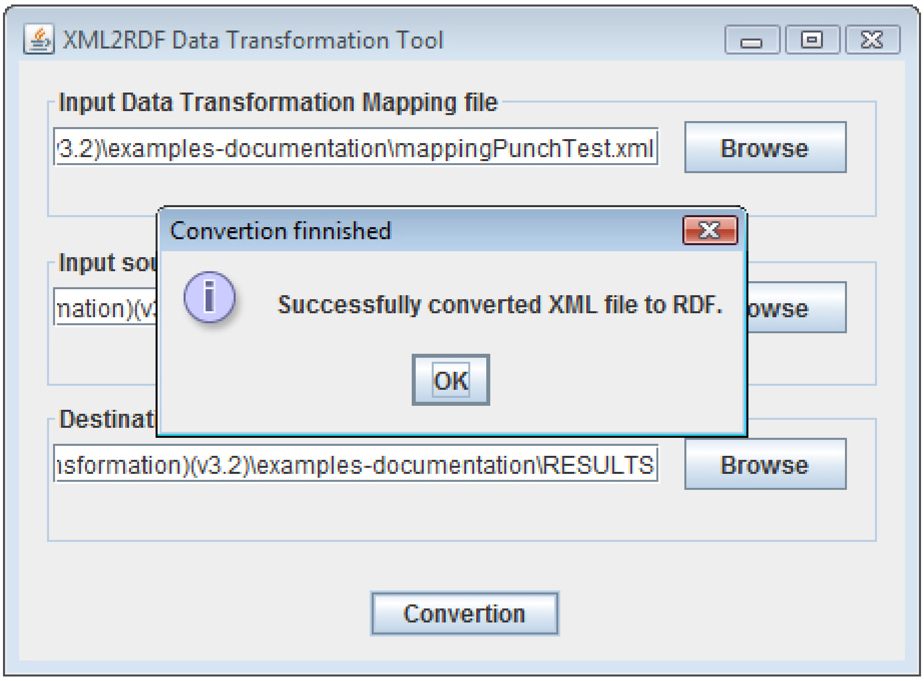
Figure2.6 XML2RDF Transformation Tool

The user imports two files in the XML2RDF Data Transformation Tool.

[53] http://www.cidoc-crm.org/tools.html

[54] Kondylakis, H., Doerr, M., Plexousakis, D., "Mapping Language for Information Integration", Technical Report 385, ICS-FORTH, 2006.

- The first file is a mapping file that describes the mapping between the xml source input file and the target schema ontology and
- the second file is the source xml file.
- The third file is the .rdf file generated by the tool.



Input Data Transformation Mapping file

3.2)\examples-documentation\mappingPunchTest.xml

Input sou

nation)(v

Destinat

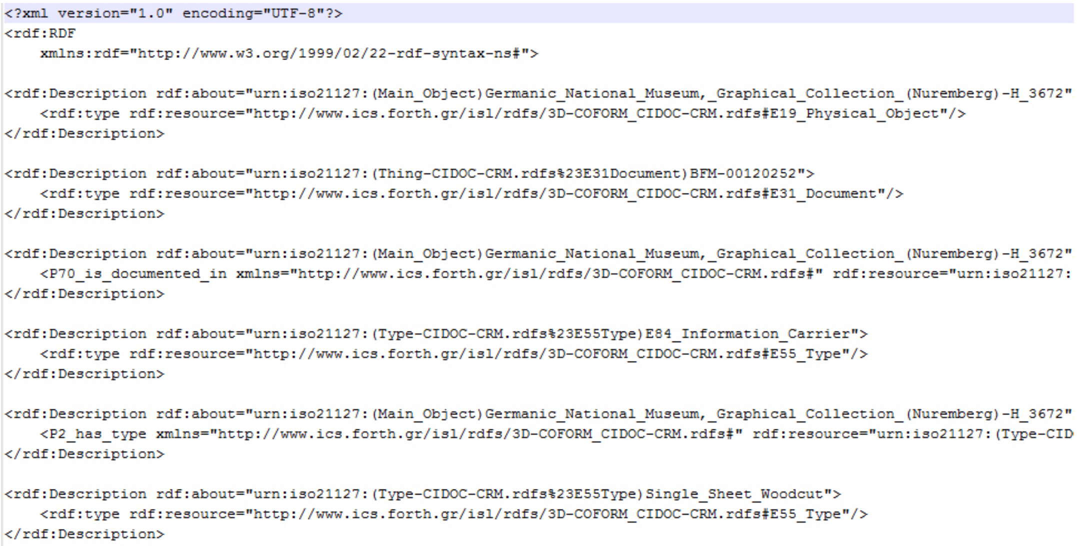
ısformation)(v3.2)\examples-documentation\RESULTS

Convertion finnished

Successfully converted XML file to RDF.

Figure 2.7: XML2RDF Transformation Tool (success message).

When the transformation has been concluded, a new ".rdf" file is created containing the conversion from xml to rdf. An example is given below:



```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<rdf:Description rdf:about="urn:iso21127:(Main_Object)Germanic_National_Museum,_Graphical_Collection_(Nuremberg)-H_3672"
    <rdf:type rdf:resource="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#E19_Physical_Object"/>
</rdf:Description>

<rdf:Description rdf:about="urn:iso21127:(Thing-CIDOC-CRM.rdfs%23E31Document)BFM-00120252">
    <rdf:type rdf:resource="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#E31_Document"/>
</rdf:Description>

<rdf:Description rdf:about="urn:iso21127:(Main_Object)Germanic_National_Museum,_Graphical_Collection_(Nuremberg)-H_3672"
    <P70_is_documented_in xmlns="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#" rdf:resource="urn:iso21127:
</rdf:Description>

<rdf:Description rdf:about="urn:iso21127:(Type-CIDOC-CRM.rdfs%23E55Type)E84_Information_Carrier">
    <rdf:type rdf:resource="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#E55_Type"/>
</rdf:Description>

<rdf:Description rdf:about="urn:iso21127:(Main_Object)Germanic_National_Museum,_Graphical_Collection_(Nuremberg)-H_3672"
    <P2_has_type xmlns="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#" rdf:resource="urn:iso21127:(Type-CID
</rdf:Description>

<rdf:Description rdf:about="urn:iso21127:(Type-CIDOC-CRM.rdfs%23E55Type)Single_Sheet_Woodcut">
    <rdf:type rdf:resource="http://www.ics.forth.gr/isl/rdfs/3D-COFORM_CIDOC-CRM.rdfs#E55_Type"/>
</rdf:Description>
```

Figure 2.8: XML2RDF Transformation Tool (results.rdf file).

2.5.5 KARMA

Karma (open source tool – Apache 2 license) facilitates users to integrate data (e.g. databases, delimited text files, XML, JSON, KML and Web APIS) by automatically modeling them according to an ontology class of their choice with the help of a graphical user interface. Karma recognizes the ontology and generates a model to tie together the classes. Data of different formats can be normalized and restructured by the user. By the completion of the model, the integrated data is published as RDF and stored in a database[55].

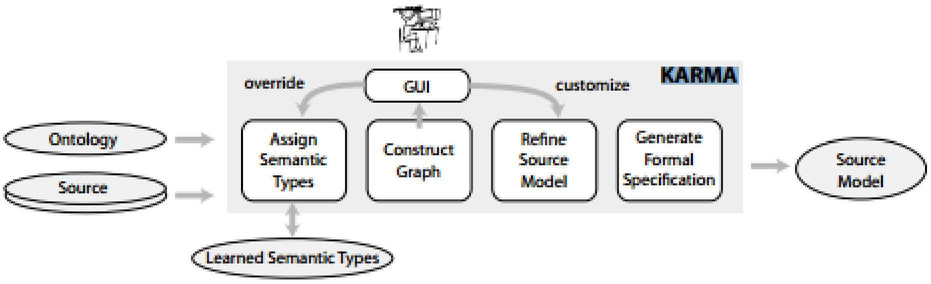


Figure 2.9: Architecture of KARMA *(figure available at http://www.isi.edu/~ambite/eswc-karma.pdf)*

The input to the system[56] is an

1. OWL ontology
2. Data sources that the user wants to map the ontology
3. Semantic types already recognizable by the system

Output of the system is

1. Model specifying the mapping between the ontology and the source
2. Refined database of semantic types

Steps of modeling process:

Assign Semantic Types: The system based on data values in each column maps sources to a node in the ontology.

[55] Karma – A Data Integration Tool, available online at http://www.isi.edu/integration/karma/

[56] Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P., "Semi-Automatically Mapping Structured Sources into the Semantic Web", The Semantic Web: Research and Applications, vol 7295, pp. 375-390.Springer, 2012.

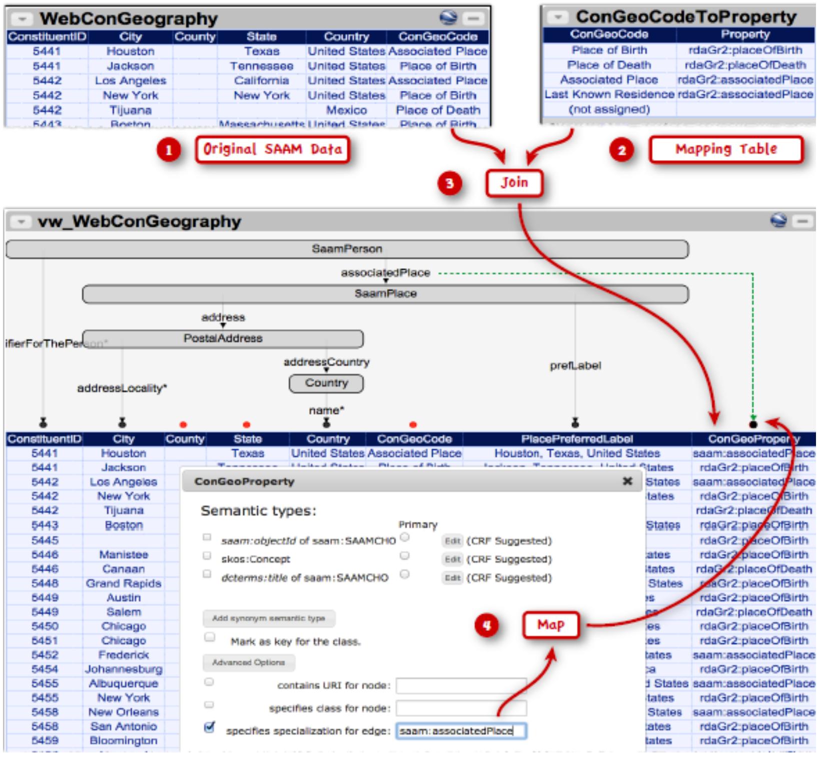
Construct Graph: A graph defining the space of possible mappings. The nodes correspond to classes and the edges correspond to properties in the ontology.

Refine Source Model: The graph is updated based on user input and the mappings are developed using a Steiner Tree Algorithm.

Generate Formal Specification: A formal specification of the source model from the Steiner Tree is generated.

Karma was used by the Smithsonian American Art Museum to map metadata for 41.000 objects. SAAM includes 407 classes and 105 data properties[57]

Karma provides a user interface to let users assign semantic types to the columns of a data source



WebConGeography

ConstituentID	City	County	State	Country	ConGeoCode
5441	Houston		Texas	United States	Associated Place
5441	Jackson		Tennessee	United States	Place of Birth
5442	Los Angeles		California	United States	Associated Place
5442	New York		New York	United States	Place of Birth
5442	Tijuana			Mexico	Place of Death
5443	Boston		Massachusetts	United States	Place of Birth

ConGeoCodeToProperty

ConGeoCode	Property
Place of Birth	rdaGr2:placeOfBirth
Place of Death	rdaGr2:placeOfDeath
Associated Place	rdaGr2:associatedPlace
Last Known Residence	rdaGr2:associatedPlace
(not assigned)	

ConstituentID	City	County	State	Country	ConGeoCode	PlacePreferredLabel	ConGeoProperty
5441	Houston		Texas	United States	Associated Place	Houston, Texas, United States	saam:associatedPlace
5441	Jackson						rdaGr2:placeOfBirth
5442	Los Angeles						saam:associatedPlace
5442	New York						rdaGr2:placeOfBirth
5442	Tijuana						rdaGr2:placeOfDeath
5443	Boston						rdaGr2:placeOfBirth
5445							rdaGr2:placeOfBirth
5446	Manistee						rdaGr2:placeOfBirth
5446	Canaan						rdaGr2:placeOfDeath
5448	Grand Rapids						rdaGr2:placeOfBirth
5449	Austin						rdaGr2:placeOfBirth
5449	Salem						rdaGr2:placeOfDeath
5450	Chicago						rdaGr2:placeOfBirth
5451	Chicago						rdaGr2:placeOfBirth
5452	Frederick						saam:associatedPlace
5454	Johannesburg						rdaGr2:placeOfBirth
5455	Albuquerque						saam:associatedPlace
5455	New York						rdaGr2:placeOfBirth
5458	New Orleans						saam:associatedPlace
5458	San Antonio						rdaGr2:placeOfBirth
5459	Bloomington						rdaGr2:placeOfBirth

ConGeoProperty

Semantic types:

		Primary	
□	*saam:objectId* of saam:SAAMCHO	○	Edit (CRF Suggested)
□	skos:Concept	○	Edit (CRF Suggested)
□	*dcterms:title* of saam:SAAMCHO	○	Edit (CRF Suggested)

Add synonym semantic type

□ Mark as key for the class.

Advanced Options

□ contains URI for node:

□ specifies class for node:

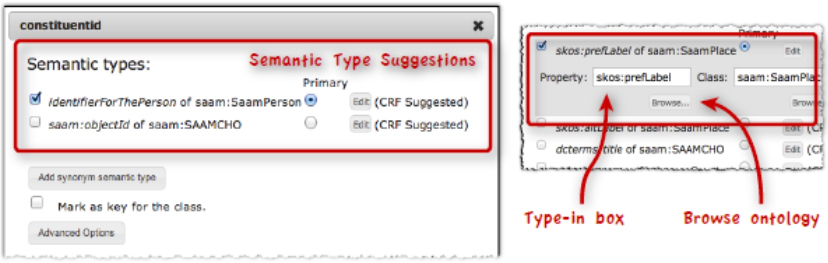
☑ specifies specialization for edge: saam:associatedPlace

Figure 2.10: Graphical User Interface (GUI) of KARMA *(figure available at http://eswc-conferences.org/sites/default/files/papers2013/szekely.pdf)*

[57] Szekely, P., Knoblock, C.A., Yang, F., Zhu, X., Fink, E.E., Allen, R., Goodlander,G., "Connecting the Smithsonian American Art Museum to the Linked Data Cloud", The Semantic Web: Semantics and Big Data, vol. 7882, pp. 593-607. Springer, 2013.

Also, users can see the Karma-proposed mappings and can adjust them if necessary, and by enabling users to work with example data rather than just schemas and ontologies.

Another important feature is that Karma allows users to see the proposed mappings and offers the option to work with example data and not only schemas and ontologies.



constituentid ✖

Semantic types:

Semantic Type Suggestions

Primary

☑ *identifierForThePerson* of saam:SaamPerson ● Edit (CRF Suggested)

□ *saam:objectId* of saam:SAAMCHO ○ Edit (CRF Suggested)

Add synonym semantic type

□ Mark as key for the class.

Advanced Options

Primary

☑ *skos:prefLabel* of saam:SaamPlace ● Edit

Property: skos:prefLabel Class: saam:SaamPla

Browse... Browse

○ *dcterms:title* of saam:SAAMCHO ○ Edit (CR

Type-in box

Browse ontology

Figure 2.11: Semantic type suggestions in KARMA *(figure available at http://eswc-conferences.org/sites/default/files/papers2013/szekely.pdf)*

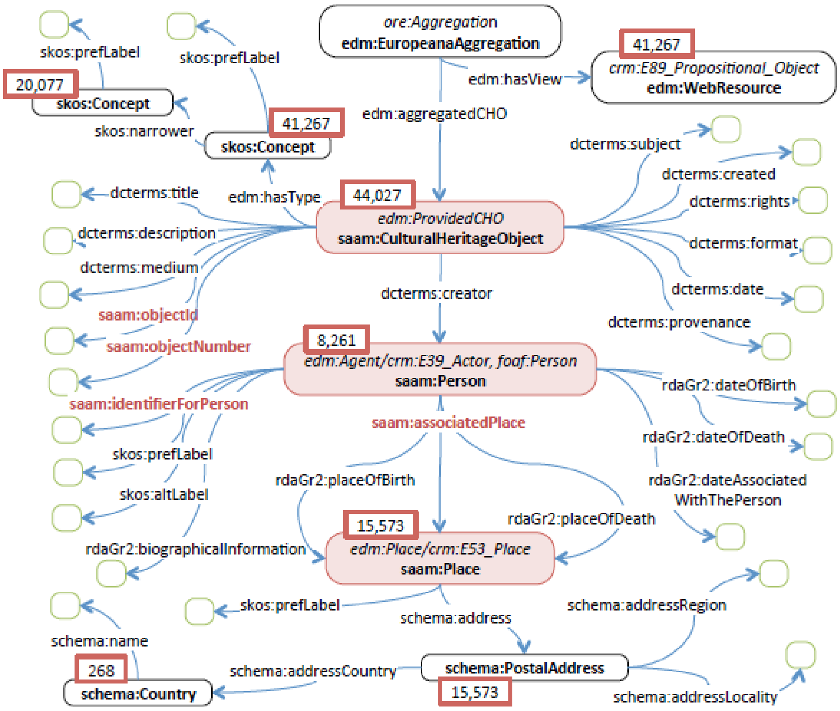


Figure 2.12: SAAM ontology *(figure available at http://eswc-conferences.org/sites/default/files/papers2013/szekely.pdf)*

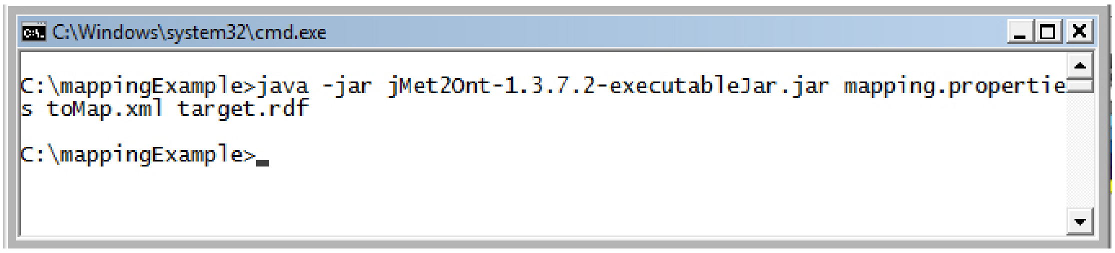
2.5.6 Jmet2ont (XML→ RDF (CIDOC))

jMet2ont is an open source tool mapping XML based metadata (e.g. Dublin Core, MARC/XML etc) to ontology formats[58,59,60] and developed for the SYNAT project[61]. It generates RDF triplets consistent with the Europeana Data Model[62], the CIDOC CRM[63] for museum objects and FRBRoo for library ones. It implements Sesame[64], which is the standard network for processing RDF files. In order to use this tool, there is no need for special programming knowledge. It makes use of UTF-8 encoding.
It is available at http://fbc.pionier.net.pl/pro/jmet2ont .
A Visual mappings editor is absent. However, it is intended to be done in future work which also includes creation of graphical mapping rules definition interface, rule language syntax optimization, performance optimization, annotation and enrichment plug-in option.
In order to run this tool which is packaged in a Java exe file, we have to open the windows CMD console and insert the following command:

```
java -jar jMet2Ont-X.X-executableJar.jar properties-file source-file target-file
```

In the following example the jmet2ont – 1.3.7.2 – executable .jar is the current version name. The mapping.properties file is the path to the configuration file where you specify the mapping rules file and can provide information for the mapper. The toMap.xml is the source file containing the data which is going to be mapped. The Target.rdf file is the resulting RDF file.

[58] http://fbc.pionier.net.pl/pro/jmet2ont/
[59] http://www.cidoc-crm.org/mapping_technology/jMet2Ont.pdf
[60] Walkowska, J., Werla, M., “Advanced Automatic Mapping from Flat or Hierarchical Metadata Schemas to a Semantic Web Ontology”, Theory and Practice of Digital Libraries, Lecture Notes in Computer Science, vol. 7489, pp 260-272. Springer, 2012.
[61] “SYNAT is a Polish national research project aimed at the creation of universal open repository platform for hosting and communication of networked resources of knowledge for science, education, and open society of knowledge”. http://www.carpet-project.net/en/projects/projects-providers/project-developments/synat/
[62] http://www.europeana.eu/schemas/edm/
[63] http://www.cidoc-crm.org/
[64] http://www.openrdf.org/



```
C:\mappingExample>java -jar jMet2Ont-1.3.7.2-executableJar.jar mapping.propertie
s toMap.xml target.rdf

C:\mappingExample>_
```

Figure 2.13: Jmet2Ont mapping example

2.5.7 MINT[65,66] Tool for Mapping (Metadata Interoperability Services)

MINT is a mapping and aggregating data tool of NTUA, used to determine semantic mappings of source and target schemas. It generates an XSLT for the transformation of XML docs into other objects including other docs. XSLT is a Turing complete language, specifying computations performed by computers.

MINT has a visual mapping editor for XSL language to bridge metadata standards and to maintain interoperability with aggregators and Europeana at the same time .

It is used by the Europeana Photography project for the transformation of in house data to the standard chosen for the project and by other projects. MINT is web-based and offers many services for metadata aggregation.

It integrates data from multiple sources, maps the imported records to the intermediate metadata schema and transforms and stores the metadata in the repository.

The tool offers the following potentials to Europeana Photography project:

- Provision of metadata records in various source formats
- Conversion of metadata to the project’s standard
- Mapping local terminologies to adopted reference ones
- Submission of records to Europeana

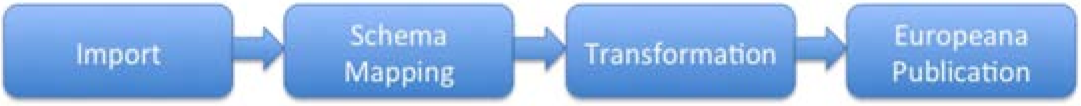
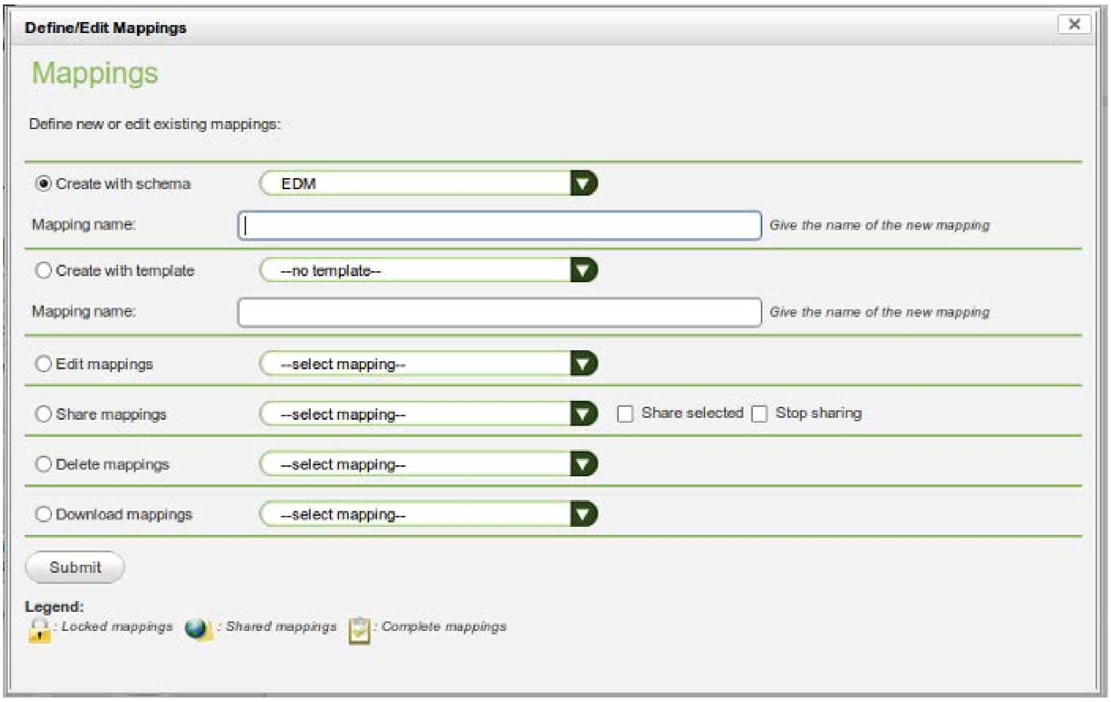


Figure 2.14: MINT overall workflow *(figure available at http://www.pro.europeana.eu/documents/1204249/0/D5.2+-+The+MINT+Mapping+Tool)*

[65] http://mint.image.ece.ntua.gr/redmine/projects/mint/wiki/Mappings

[66] http://www.pro.europeana.eu/documents/1204249/0/D5.2+-+The+MINT+Mapping+Tool The Mint Mapping Tool, Accessed on 11 December 2013



Define/Edit Mappings

Mappings

Define new or edit existing mappings:

● Create with schema — EDM

Mapping name: — *Give the name of the new mapping*

○ Create with template — --no template--

Mapping name: — *Give the name of the new mapping*

○ Edit mappings — --select mapping--

○ Share mappings — --select mapping-- □ Share selected □ Stop sharing

○ Delete mappings — --select mapping--

○ Download mappings — --select mapping--

Legend:

🔒 : *Locked mappings* 🌐 : *Shared mappings* ✓ : *Complete mappings*

Figure 2.15: MINT Graphical User Interface (*Figure available at: http://mint.image.ece.ntua.gr/redmine/projects/mint/wiki/Mappings)*

Chapter 3

3 Presentation of our proposed semi-automatic mapping tool

3.1 Introduction

In the previous section, the following mapping tools were described: AMA, Annocultor, RDFer, Karma, XML to RDF transformation tool, MINT, and jMet2Ont. For the purpose of examining their characteristics in order to proceed to a comparison among them, the three following parameters were considered: a) requirement of programmer knowledge in order to define the mappings, b) provision of a UI (User Interface) and c) provision of suggested mappings to the user.

Four of the above tools (Annocultor, RDFer, XML2RDF, jMet2Ont) require programmer knowledge in order to create the mappings since the user shall write code in a specific programming language or define rules in a specific mapping description language manually. For example, in Annocultor in order to construct a converter for a specific dataset in order to convert XML files to RDF, the user shall create a simple java program where the existing rules need to be put together. In RDFer the user has to write manually specific configuration files, in XML2RDF he shall create his own mapping files written in a specific mapping description language and in jMet2Ont the user shall write his own mapping rules based on a specific mapping description language, as well.

Concerning the second parameter, only three tools (AMA, KARMA and MINT) have a user friendly interface (UI). However, AMA and MINT mapping tools lack the third parameter mentioned above. In other words, they do not offer users the possibility to see suggested mappings, re-use them or adjust them. The only tool which satisfies all three requirements, that is, requires no programmer knowledge, has a developed user interface and gives suggested mappings to the user is KARMA tool.

Karma learns the assignment of semantic types to columns of data from user assignments by using a conditional random field (CRF) and in this way it automatically suggests types for not assigned data columns. Also, it re-trains automatically the CRF model after the manual assignments.

In our system a semi-automatic mapping tool that maps a source schema to a target schema (ontology) and has the following features, is proposed:

- ✓ Supports a friendly Graphical User Interface (GUI) that consists of the following components:
    - Source analyzer → analyzes the source schema.
    - Target schema analyzer → analyzes and visualizes the target schema ontology in order to be well understood by the user.
    - Target schema path editor that:
        - Highlights errors to the user when forming the target schema paths by defining the type of error and the position of the error in the target path. A console view is used in order to visualize the errors to the user.
        - Validates the target schema paths by checking the inheritance from classes to classes and from properties to properties according to the resource description framework schema of the target ontology (.rdfs).
        - During the target schema path formation the "Target path Suggestion" module suggests the right properties/entities for a specific position in the target path. No need for the user to remember the inheritance of the classes and properties.
        - Visualization of target path using a JList (in Java) that contains the classes and properties.

          (*The editor mentioned above is called "CRM editor" and it is analyzed in detail in the following chapters.*)
- ✓ Produces automatically mapping files between the source schema and the target schema (ontology) and stores them in the mapping memory.
    - There is no need for user to write manually and validate mapping files

  - The user does not have to be an IT expert and know the (XSD) schema of the mapping language anymore because the system automatically creates the mapping files based on the mapping language.
- ✓ Follows a reuse strategy by re-using the mappings that are stored in the mapping memory "mapping-memories"[67].
- ✓ Suggests mappings to the user by exploiting the mappings stored in the mapping memory (re-use of mapping files) with the assistance of a schema matching tool.
  - The user can either accept or deny the suggestion provided by the system.
  - User feedback is taken into consideration after the mapping process.
- ✓ Proposes a Learning method in order to improve the accuracy of the schema matching tool and reduce the false negatives matches. (false negative matches are relevant matches that were not found by the schema matching tool)
  - A training set is produced automatically by an automatic algorithm based on the data stored in the mapping memory.
  - Feedback taken out of the learning method can be used in the schema matching process by the matchers.

Our system can be used by non IT experts that may not have programming education/knowledge. It is easy to use and interact with because it provides the following features.

- ✓ It creates automatically mapping files.
- ✓ It analyzes in details the target ontology.
- ✓ It validates automatically the target ontology paths.
- ✓ It highlights errors.
- ✓ It suggests the right properties/entities for the specified ontology.

Before the development of the system, the users had to:

[67] Doerr, M., Felicetti, A., "A Reference model for Data Mapping Tools", The CIDOC Conceptual Reference Model, FORTH. 2012. Available from: http://www.cidoc-crm.org

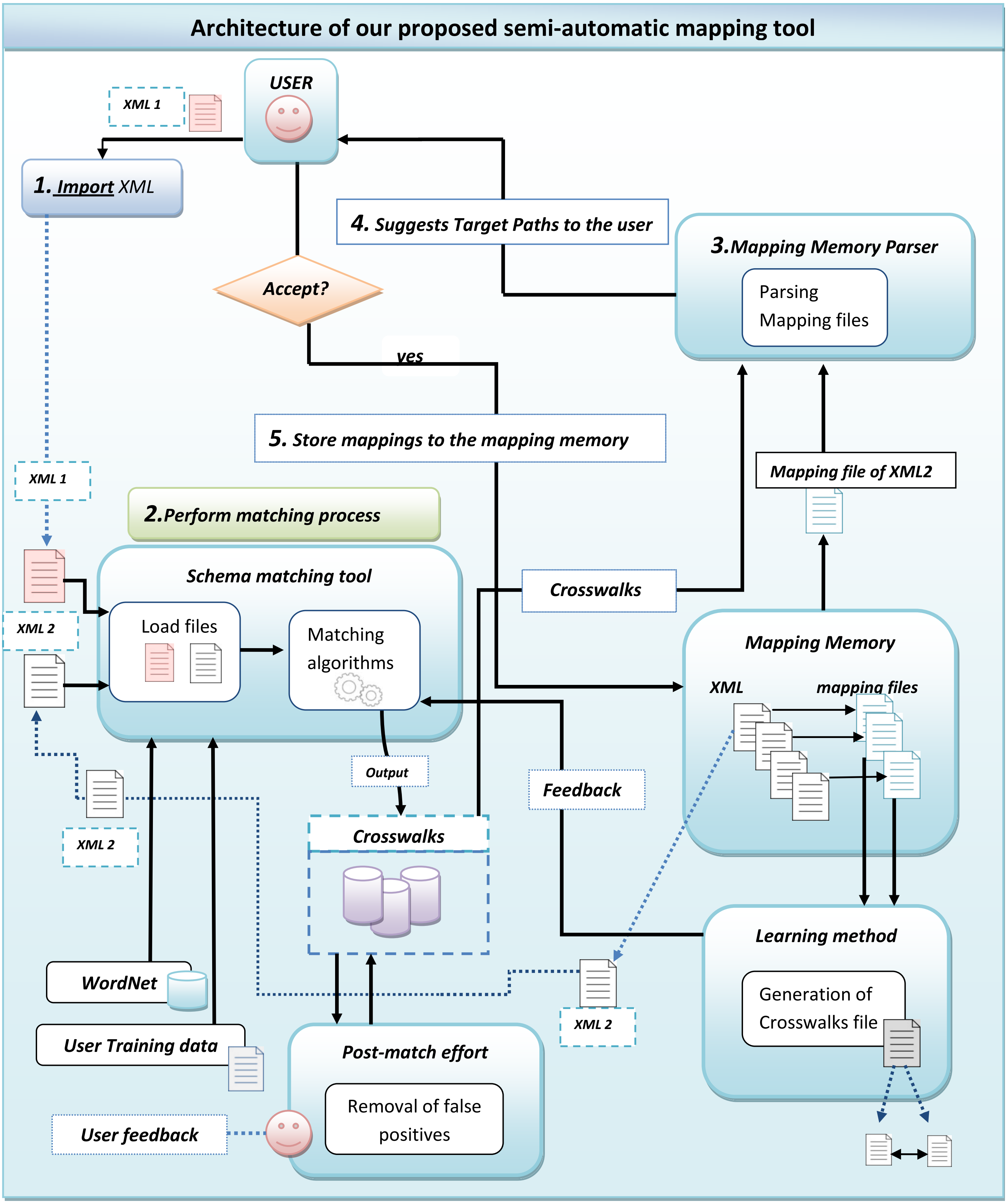
- Write manually the mapping files.
- Learn the XSD schema of the mapping language.
- Form the target paths manually by checking the inheritance of classes and properties.
- Make the validation of all the target paths.
- Check whether the XML mapping files are well-formed and correct them.

In the following sections, we analyze the architecture of our system and the way it works by describing the individual components of it, the GUI (Graphical User interface) and the Java Class Library that consists of essential modules implemented and used in our project that can be imported and used in different applications.

3.2 System's architecture

In this section we describe the architecture of our proposed semi-automatic mapping tool and how the system functions:

Figure 3.1 Architecture of our semi-automatic mapping tool



Our system functions as follows:

1. User imports the XML1 document file that he would like to map to the target schema ontology.

2. The schema matching process is performed in step 2 with the assistance of a schema matching tool.

   The schema matching tool:

   - takes as input two XML document files:
     - **XML1**
       - XML1 is the users' XML file
     - **XML2**
       - XML2 is an XML document file that comes from the mapping memory. As we already know the mapping memory contains both XML document files with their mapping files respectively.
   - It performs the schema matching process in the background between XML1 and XML2. In this process a set of algorithms is involved and internal parameters are set in order to improve the accuracy of it. (*More information for the schema matching is given in the 'Schema matching process' section later on in this chapter*).
   - An external thesauri (WordNet) [68]and user training data are loaded onto the tool in order to improve its results.
   - The schema matching tool stores the crosswalks between the two XML document files in the database.

3. After the matching process, the user can additionally remove manually the irrelevant matches[69], the false positives from the final match result.

[68] http://wordnet.princeton.edu/

[69] Duchateau, F., Bellahsene, Z., Coletta, R., "A Flexible Approach for Planning Schema Matching Algorithms", On the Move to Meaningful Internet Systems: OTM 2008, vol. 5331, pp. 249-264. Springer, 2008.

This step is optional since the post-match effort varies heavily with the background knowledge and the abilities of the users.[70]

4. The mapping memory parser component parses the mapping file of XML2 from mapping memory. As we already know the mapping memory consists of already mapped XML document files with their mappings files. These mappings files describe the mappings between XML and the target ontology (e.g. CIDOC-CRM). So, the fore mentioned component outputs a list of suggested target paths to the user based on the crosswalks found by the schema matching process.

5. The User can either accept or deny the suggested target path and feedback is stored for further use.

6. The Learning method component is analyzed in detail in the following section "Learning method" and helps the Recall of the schema matching results to improve.

3.3 Schema matching process

[71]Schema matching is the generation of semantic correspondences among schemas in an effort to find correspondences among elements in two schemas.[72]The schema matching process can't be fully automated. It needs user feedback, because the same element in a schema can refer to different entities. Examples are given below:

- An element titled <area> can refer to the

[70] Do, H.H., Melnik, S., Rahm, E., "Comparison of Schema Matching Evaluations", Web, Web-Services, and Database Systems, vol. 2593, pp. 221 – 237. Springer, 2003.

[71] Smith, K., Mork, P., Seligman, L., Rosenthal, A., Morse, M., Wolf, C., Allen, D., and Li, M., "The role of schema matching in large enterprises", In: Proceedings of the 4th Biennial Conference on Innovative Data Systems Research (CIDR), 2009.
Available from: http://arxiv.org/ftp/arxiv/papers/0909/0909.1771.pdf

[72] Knoblock, C., "Schema matching", Available from: http://www.isi.edu/integration/courses/csci548_2008/slides08/Matching.pdf, 2008, Accessed 10 March 2013.

1. location of the house or
2. square-feet area of the house.

- An element titled <name> can refer to an
    1. actor's name or
    2. to a place name.

Also, elements with different names may refer to exactly the same entity. For instance <area> and <address> may refer to the location of a house.

Manually specifying schema matches is a time consuming task[73] and for that reason we use a semi-automatic schema matching tool that automates the schema matching process and takes the user's feedback into consideration (post-match effort). We have tuned the schema matching tool in order to have better quality in our results. More detailed information is included in the Evaluation section of the thesis.

We have used and expanded the schema matching tool developed in the previous master theses[74,75,76]. The expanded schema matching tool provides a set of match algorithms and combines linguistic and structural matching methods. The inputs of the tool are XML document files and the output is a set of correspondences between those files.

[73] Li, Y., Liu, D., Zhang,W.,"A Generic Algorithm for Heterogeneous Schema Matching", International Journal of Information Technology, pp. 36-43, 2005.

[74] Manakanatas, D., "Design and Implementation of a tool for semi-automated semantic schema matching", Master of Science Thesis Computer Science Department, University of Crete", E-locus University of Crete Institutional Repository, February 2006. Available from: http://elocus.lib.uoc.gr/dlib/c/5/7/metadata-dlib-2006manakanatas.tkl

[75] Kalaitzaki, M., : "Design and implementation of a system for semantic schema matching. Master of Science Thesis Computer Science Department, University of Crete", 2009. Available from: http://elocus.lib.uoc.gr/dlib/6/5/5/metadata-dlib-4d555a739dd11f5df7c7801d0be78ef8_1275560441.tkl

[76] Daskalaki, E., "Development and experimental evaluation of an ontology to ontology schema & instance matching system, Master of Science Thesis Computer Science Department, University of Crete", E-locus University of Crete Institutional Repository, 2011, Available from: http://elocus.lib.uoc.gr/dlib/d/d/b/metadata-dlib-1322814460-437568-20114.tkl

3.4 Mapping memory

The Mapping Memory is a repository that stores mapping files that come from different partners belonging to different domains: Culture heritage domain, digital libraries domain, archives domain, biomedical domain and others.

The **Mapping Memory** plays multiple roles in our system since we exploit its data for three different reasons:

1) Stores the mapping files that contain the mapping memories which are used for the suggestion of the target paths to the user. (re-use strategy)
2) Exploits information for the **Learning Component algorithm** by giving feedback and improving the results of the Schema matching tool A. Creation of a dictionary of terms.
3) Acts as a **second** schema matching tool (B) that produces/outputs crosswalks between source schemas.

3.5 Learning method

3.5.1 Introduction

In order to improve the accuracy of the schema matching tool, we used a matching strategy and we implemented a Learning method that improves the results of the matching process.

[77]Machine learning used by matchers operates in two phases.

- Learning or training phase
- Matching phase

Training phase: In this phase, training data for the learning process is created. I implemented an automatic algorithm that produces the training set by exploiting the data stored in the mapping memory and additionally generates crosswalks

[77] Shvaiko, P., Euzenat, J., “Learning methods” .Chapter In: *Ontology Matching*. pp. 133. Springer, 2007. http://wtlab.um.ac.ir/images/e-library/ontology/Ontology%20Matching.pdf

between the two metadata source files[78].This extraction knowledge algorithm is analyzed in detail in the next section. We chose to gather this training set from the mapping memory since the mapping files that come from different partners contain a lot of information (both for the source and target schema). Also, the amount of mapping files in the mapping memory is going to increase on a daily basis and our system is going to continuously learn from them.

In the second phase we run the learning algorithm on the training set that we have already gathered.[79] In the evaluation section we are going to see how the learning method affects the accuracy of our system with other adjustment parameters.

[78] Gaitanou, P., Bountouri, L., Gergatsoulis, M., "Automatic Generation of Crosswalks through CIDOC CRM.", Metadata and Semantics Research, vol. 343, pp. 264-275. Springer, 2012.

[79] http://en.wikipedia.org/wiki/Supervised_learning

3.5.2 Architecture

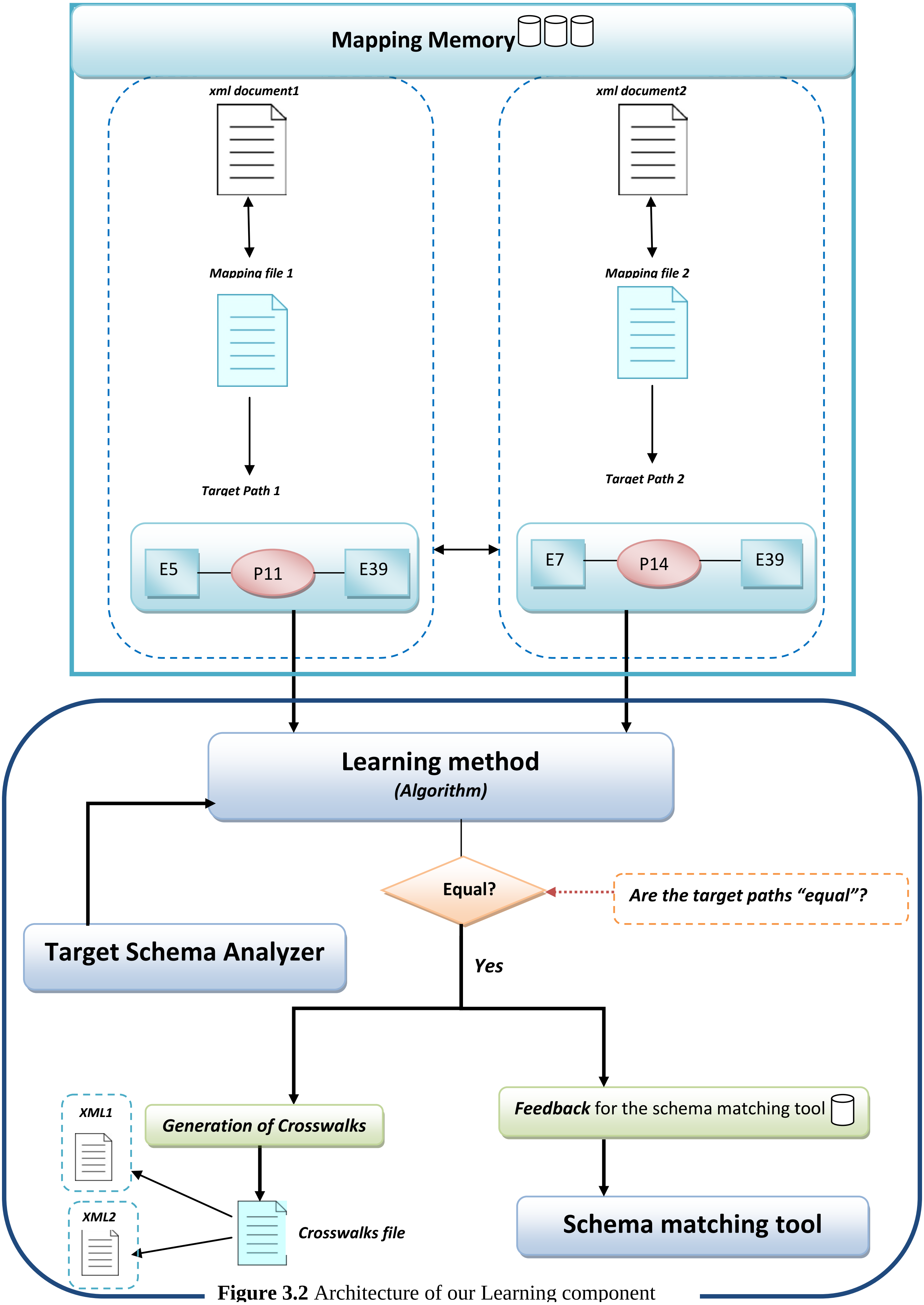


Figure 3.2 Architecture of our Learning component

In this section we describe the architecture of our Learning method.

As we have already mentioned the mapping memory plays three different roles:

- Re-uses mappings in order to improve the mapping process and suggest target paths to the user.
- Implements an automatic Learning method algorithm that produces the crosswalks between metadata schemas (schema-matching process).
- Gives feedback to the schema matching tool, dictionary of terms in order to improve the schema marching accuracy.

In order to perform the comparison of the mapping files, the mapping files shall belong to the same target ontology or to an extension of it. That's why the "Target Schema Analyzer" is involved in this procedure.
(The Target Schema Analyzer is analyzed in detail in the "Java Class Library" section).

3.5.3 Algorithm

In this section we analyze the algorithm used by the Learning method in order to create the training set. This algorithm[80] automatically generates crosswalks between XML metadata schemas using the mappings defined between the schemas and the target schema (e.g. CIDOC CRM) and gives feedback to the mapping memory component.

In the implementation of the algorithm we

- Load a pair of mapping files (expressed in a mapping description language[81]) that contain the mappings of a metadata schema to the target schema ontology e.g. CIDOC-CRM.
- We parse the mapping files in order to find *"equal"* target paths between the two mapping files. (*We explain later what we mean by saying "equal" paths*)

[80] Gaitanou, P., Bountouri, L., Gergatsoulis, M., "Automatic Generation of Crosswalks through CIDOC CRM.", Metadata and Semantics Research, vol. 343, pp. 264-275. Springer, 2012.

[81] Kondylakis, H., Doerr, M., Plexousakis, D., "Mapping Language for Information Integration", Technical Report 385, ICS-FORTH, 2006.

- If the target paths are *“equal”* we
    - generate a crosswalks file containing the crosswalks (the source schema paths) between the metadata schemas
    - send feedback to the schema matching tool in order to improve its accuracy and results for future schema matches between metadata schemas.
- If the target paths are *“not equal”*, no crosswalk entry is added to the crosswalks file and no feedback is given to the schema matching tool.

In the algorithm I use a Java module that I have implemented and it is called ***Target Schema Analyzer***. This Java module analyzes the target schema ontology and provides a list of useful and essential methods that are used further in the learning method algorithm for the calculation of the “*equality*” of the target paths.
The ***Target Schema Analyzer*** is analyzed in detail in the following sections.

“Definition 1.[82] *A* CIDOC CRM path *is a sequence of the form:* $C0 \rightarrow P1 \rightarrow C1 \rightarrow . . . \rightarrow Pn \rightarrow Cn$ *with* $n \geq 0$, *such that* Ci, *with* $0 \leq i \leq n$, *are CIDOC CRM classes and* Pi, *with* $1 \leq i \leq n$, *are CIDOC CRM properties.”*

“Definition 2. [83]*Let A,B be two CIDOC CRM paths where A is of the form* $C0 \rightarrow P1 \rightarrow C1 \rightarrow . . . \rightarrow Pn \rightarrow Cn$, *and B is of the form* $C0' \rightarrow P1' \rightarrow C1' \rightarrow . . . \rightarrow Pn' \rightarrow Cn'$, *with* $n \geq 0$. *We say that A* Isa-subsumes *B if for each i with* $0 \leq i \leq n$, *Ci is either the same class or a subclass of Ci’ and for each j with* $1 \leq j \leq n$, *Pj is either the same property or a subproperty of Pj′.”*

<u>Example 1</u>. We have the following target paths (TP stands for Target Path)

TP1: E5_Event → P11_had_participant → E39_Actor
*TP2:*E7_Activity → P11_had_participant → E21_Person

“E7_Activity” is a subclass of “E5_Event”.

[82] Gaitanou, P., Bountouri, L., Gergatsoulis, M., “Automatic Generation of Crosswalks through CIDOC CRM.”, Metadata and Semantics Research, vol. 343, pp. 264-275. Springer, 2012.
[83] Ibid.

“P11_had_participant” is the same property as “P11_had_participant”.
“E21_Person is a subclass of “E39_Actor”.

According to definition 2 TP2 Isa-subsumes *TP1 or the target paths are “equal” because they express the same meaning.*

Example 2. We have the following target paths (TP stands for Target Path)

TP1*:*
E39_Actor → P92i_was_brought_into_existence_by → E67_Birth →
P4_has_time-span → E52_Time-Span

TP2:
E21_Person → P98i_was_born → E67_Birth → P4_has_time-span →
E52_Time Span

“E21_Person” is a subclass of “E39_Actor”
“P98i_was_born” is a subproperty of “P92i_was_brought_into_existence_by”

According to definition 2 TP2 Isa-subsumes *TP1.*
In example 2 we refer to the birth event of an actor. We express the same information by using different properties and classes in the target schema ontology.

Example 3. We have the following target paths (TP stands for Target Path)

TP1: E5_Event → P11_had_participant → E39_Actor
*TP2:*E79_Part_Addition → P111_added → E18_Physical_Thing

“E79_Part_Addtion” is a subclass of “E5_Event”.
“P111_added” is not a subproperty o “P11_had_participant”.

According to definition 2 the target paths are **not** *“equal”!*
Since we explained what we mean with “equality” of target paths, we analyze the algorithms in detail.

DESCRIPTION OF ALGORITHM

Input:	A pair of mapping files **(MF1,MF2)** ***MF** stands for "Mapping File"*
Outputs:	**1.** Crosswalks file
	2. File containing feedback for the schema matching tool.

PSEUDOCODE

```
Load Mapping Files(MF1,MF2);      // parse MF1,MF2
for each pair(S,T) in the  MF1      {  //S: Source path, T: Target path
    for each pair (S',T') in the MF2   {
        if ( T isa-subsumes T') {
            WriteToCrosswalksFile(S,S');
            StoreFeedbackForSchemaMatchingTool();
            //creation of a dictionary of terms
            SendFeedbackToSchemaMatching Tool( );
        }
    }
}
```

Table 3.1 Description of our learning method algorithm

The algorithm is analyzed as follows:

For each pair (*S*,*T*) [S: Source Path T: Target Path] in the *Mapping File 1* the algorithm checks if there is a pair (*S'*,*T'*) in the *Mapping File 2* such that the target path *T I*sa-subsumes *T'*. If such a pair exists, then the pair (*S*,*S'*) is added to the crosswalks file and feedback is sent to the schema matching tool. Otherwise, no pair is added to the crosswalks file.

We can execute the algorithm for all the mapping files in the mapping memory and the algorithm becomes:

DESCRIPTION OF EXTENDED ALGORITHM	
Input:	A repository containing Mapping Files **MFi** (0≤i≤n). ***Mapping Memory***
Outputs:	**1.** Crosswalks file for each pair (MFi ,MFj) ***MFi*** *(0≤i≤n)* ***MFj*** *(0≤j≤n)*
	2. File containing feedback for the schema matching tool for each pair**(MFi, MFj)**

PSEUDOCODE

```
for each Mapping File MFi in the Mapping Memory  {
    Load Mapping Files(MFi, MFj);       // parse MFi,MFj
    for each pair(S,T) in theMFi       {  //S: Source path, T: Target path
        for each pair (S',T') in theMFj   {
            if ( Tisa-subsumes T') {
                WriteToCrosswalksFile(S,S' , CrosswalksFile);
                StoreFeedbackForSchemaMatchingTool();
                  //creation of a dictionary of terms
                SendFeedbackToSchemaMatching Tool( );
            }
        }
    }
}
```

Table 3.2: Description of our extended learning method

3.5.4 Example

We execute the above algorithm and we give as input a pair of mapping files.

- The **mapping file 1** in Figure 3.3 contains the mappings between an XML file encoded in LIDO[84] and the target schema CIDOC-CRM.

- The **mapping file 2** in Figure 3.4 contains the mappings between an XML file encoded in VRA[85] and the target schema CIDOC-CRM

In our example we have the following target paths:

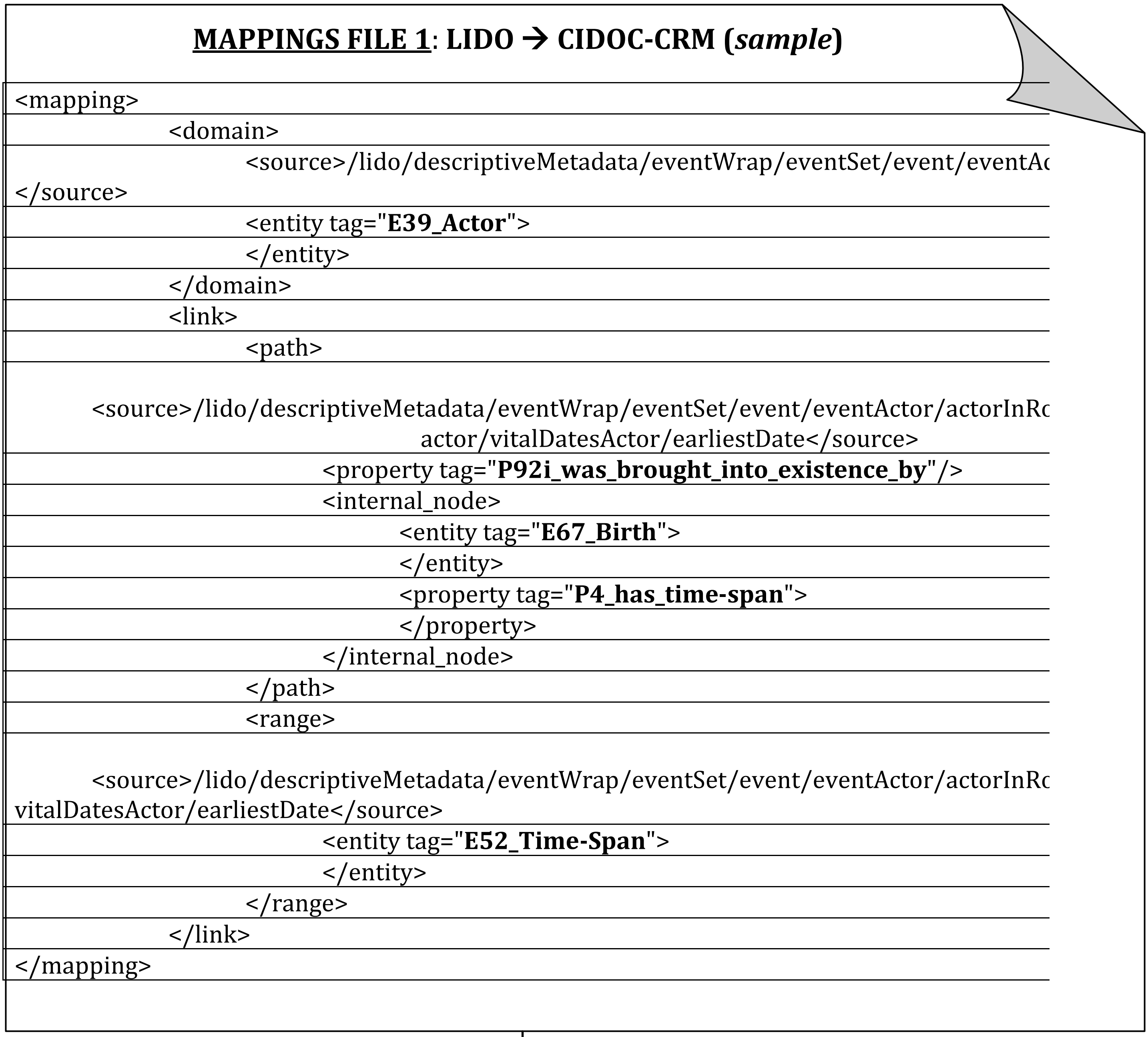
- ***Target Path1* in Mapping file 1** ***(LIDO→CIDOC-CRM)****:*
  E39_Actor → P92i_was_brought_into_existence_by → E67_Birth → P4_has_time-span → E52_Time-Span

- ***Target Path2* in Mapping file 2** ***(VRA→CIDOC-CRM):***
  E21_Person → P98i_was_born → E67_Birth → P4_has_time-span → E52_Time Span

We execute the algorithm and we find out that the *TP2* Isa-subsumes *TP1*. In that case we add the (S, S') to the crosswalks file shown in Table 3.5.

[84] http://www.lido-schema.org/schema/v1.0/lido-v1.0.xsd

[85] http://www.loc.gov/standards/vracore/



MAPPINGS FILE 1: LIDO → CIDOC-CRM (*sample*)

```
<mapping>
        <domain>
            <source>/lido/descriptiveMetadata/eventWrap/eventSet/event/eventAc
</source>
            <entity tag="E39_Actor">
            </entity>
        </domain>
        <link>
            <path>

    <source>/lido/descriptiveMetadata/eventWrap/eventSet/event/eventActor/actorInRc
                         actor/vitalDatesActor/earliestDate</source>
                <property tag="P92i_was_brought_into_existence_by"/>
                <internal_node>
                    <entity tag="E67_Birth">
                    </entity>
                    <property tag="P4_has_time-span">
                    </property>
                </internal_node>
            </path>
            <range>

    <source>/lido/descriptiveMetadata/eventWrap/eventSet/event/eventActor/actorInRc
vitalDatesActor/earliestDate</source>
                <entity tag="E52_Time-Span">
                </entity>
            </range>
        </link>
</mapping>
```

Table 3.3 Sample mapping file (LIDO→ CIDOC-CRM).

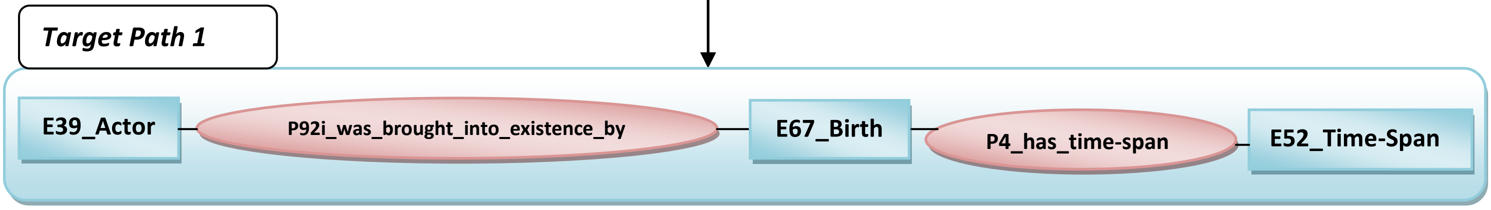
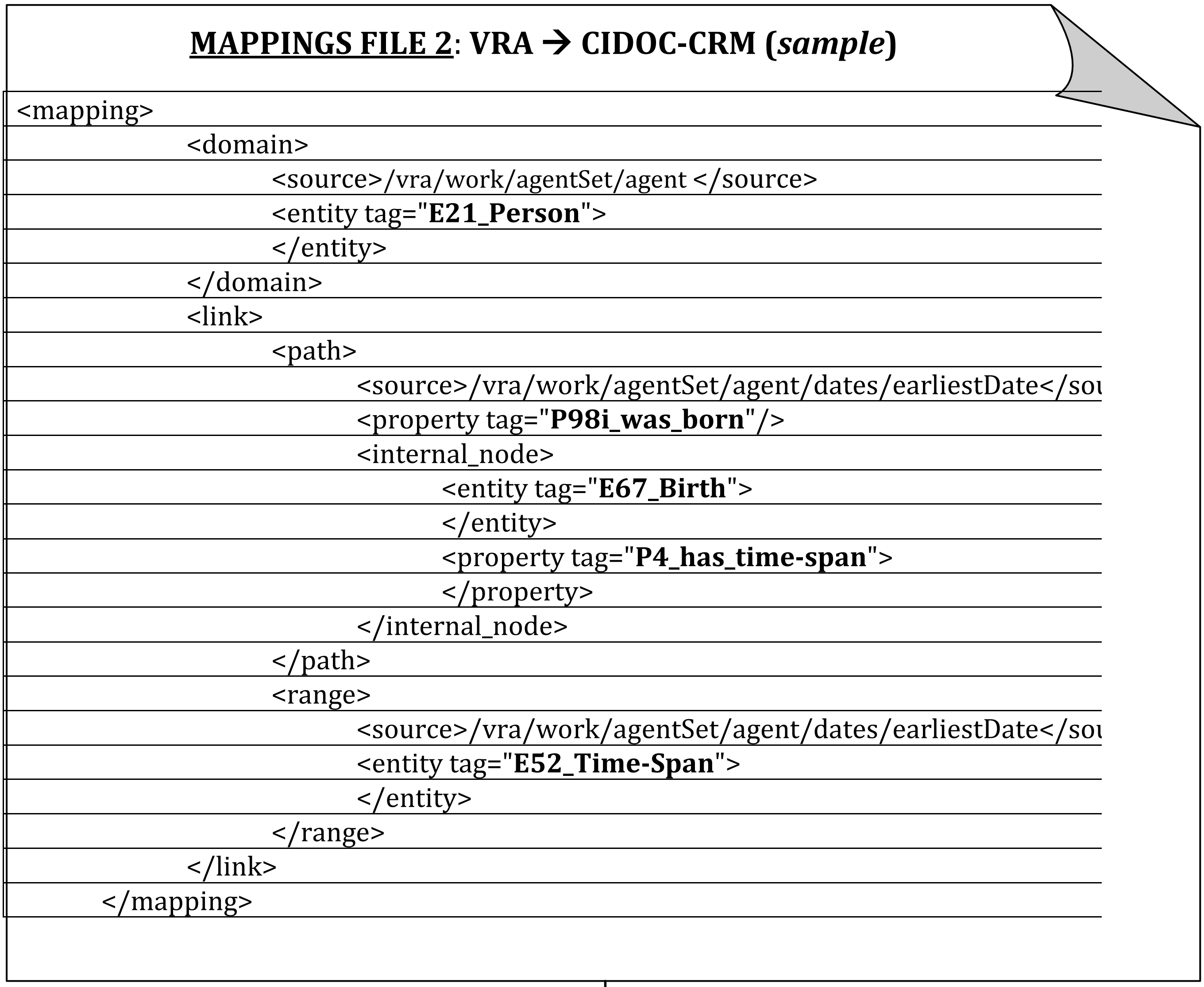


Figure 3.3 CIDOC target path for birth event (1)



MAPPINGS FILE 2: VRA → CIDOC-CRM (*sample*)

```
<mapping>
            <domain>
                  <source>/vra/work/agentSet/agent </source>
                  <entity tag="E21_Person">
                  </entity>
            </domain>
            <link>
                  <path>
                        <source>/vra/work/agentSet/agent/dates/earliestDate</sou
                        <property tag="P98i_was_born"/>
                        <internal_node>
                              <entity tag="E67_Birth">
                              </entity>
                              <property tag="P4_has_time-span">
                              </property>
                        </internal_node>
                  </path>
                  <range>
                        <source>/vra/work/agentSet/agent/dates/earliestDate</sou
                        <entity tag="E52_Time-Span">
                        </entity>
                  </range>
            </link>
      </mapping>
```

Table 3.4 Sample mapping file (VRA→ CIDOC-CRM).

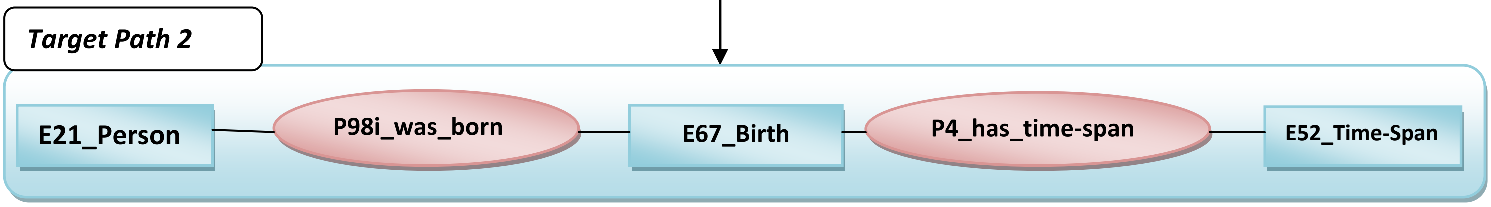
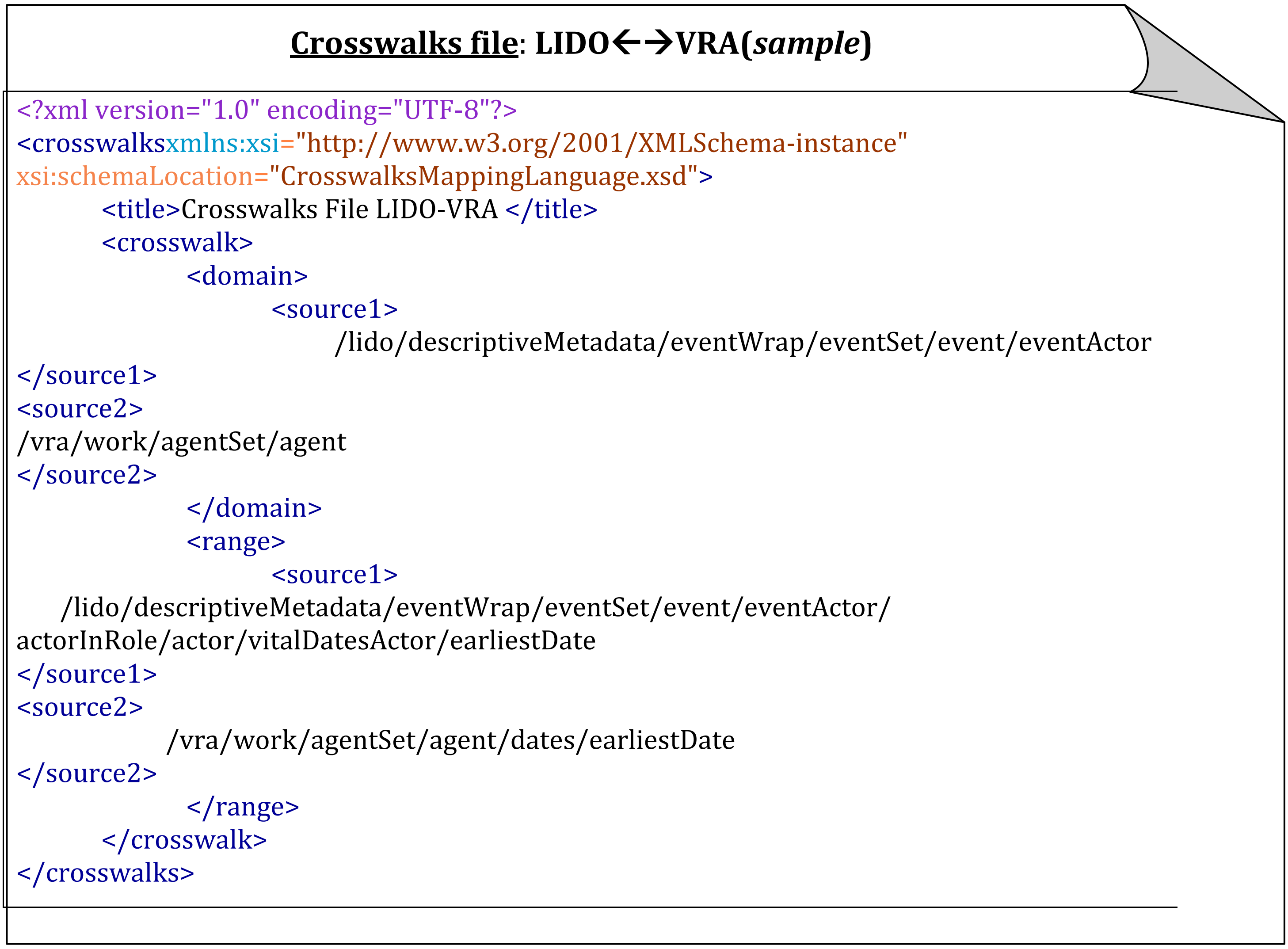


Figure 3.4 CIDOC target path for birth event (2)

Since the ***Target Path 2*** Isa-subsumes *Target Path 1*, we add to the crosswalks file the source paths of each XML file (LIDO-VRA) and we send the feedback to the schema matching tool.



Crosswalks file: **LIDO←→VRA(*sample*)**

```
<?xml version="1.0" encoding="UTF-8"?>
<crosswalksxmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="CrosswalksMappingLanguage.xsd">
        <title>Crosswalks File LIDO-VRA </title>
        <crosswalk>
                <domain>
                        <source1>
                              /lido/descriptiveMetadata/eventWrap/eventSet/event/eventActor
</source1>
<source2>
/vra/work/agentSet/agent
</source2>
                </domain>
                <range>
                        <source1>
    /lido/descriptiveMetadata/eventWrap/eventSet/event/eventActor/
actorInRole/actor/vitalDatesActor/earliestDate
</source1>
<source2>
              /vra/work/agentSet/agent/dates/earliestDate
</source2>
                </range>
        </crosswalk>
</crosswalks>
```

Table 3.5 Crosswalks file for LIDO→VRA

We present the XSD schema of ***CrosswalksMappingLanguage.xsd*** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schemaxmlns:xs="http://www.w3.org/2001/XMLSchema"elementFormDefault="
qualified">
<xs:element name="crosswalks">
<xs:complexType>
<xs:sequence>
<xs:element name="title" type="xs:string"/>
<xs:element ref="crosswalk"maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="crosswalk">
<xs:complexType>
<xs:sequence>
<xs:element ref="domain"/>
<xs:element ref="range"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="domain">
<xs:complexType>
<xs:sequence>
<xs:element name="source1" type="xs:string"/>
<xs:element name="source2" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="range">
<xs:complexType>
<xs:sequence>
<xs:element name="source1" type="xs:string"/>
<xs:element name="source2" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Table 3.6 XSD schema for describing crosswalks between two XML schemas.

3.6 Graphical User Interface (GUI)

In this section we analyze the GUI that is used in our system. The Graphical User Interface is written in Java and consists of the Source Analyzer (a component that analyzes the source schema) and the target path editor. We have integrated our Java modules in an open source tool called “SIP-Creator” by Delving[86].The SIP-Creator application imports XML files regardless of their structure or schema. With the data imported the SIP-Creator quickly analyzes the data and provides a wealth of statistical analysis in the form of histograms and graphs, maps the data to a desired target schema (xsd),validates the mapping and uploads it to the Culture-Hub (a platform that brings the data online and makes it come alive via a set of rich APIs.)[87] .

So in our system we use the Source Analyzer of the SIP-creator tool in order to analyze the imported user’s XML document file and produce statistics of the imported schema. After the user clicks on a specific element on the tree structure of the XML document, we get the full path of that element on the tree structure and we call our implemented Java modules that are analyzed further in the next section called “Java Class library”. We have also created a new Java frame named ”CRM editor” that improves the mapping process with the additional essential features that it offers. Finally, we integrated the “CRM editor” in the SIP-creator tool.

We describe with a list of figures the layout of the GUI and the steps that the user has to follow in order to map his source file to the target ontology.

[86] “Delving SIP-Creator”, Delving open-source solutions, Available from: http://www.delving.eu/the-delving-platform/sip-creator.Accessed: 1st October 2013.

[87] “Delving Culture-hub”, Delving open-source solutions, Available from: http://www.delving.eu/the-delving-platform/culture-hub. Accessed 1st October 2013.

Figure 3.5 Delving SIP-Creator Source Analyzer View [1]

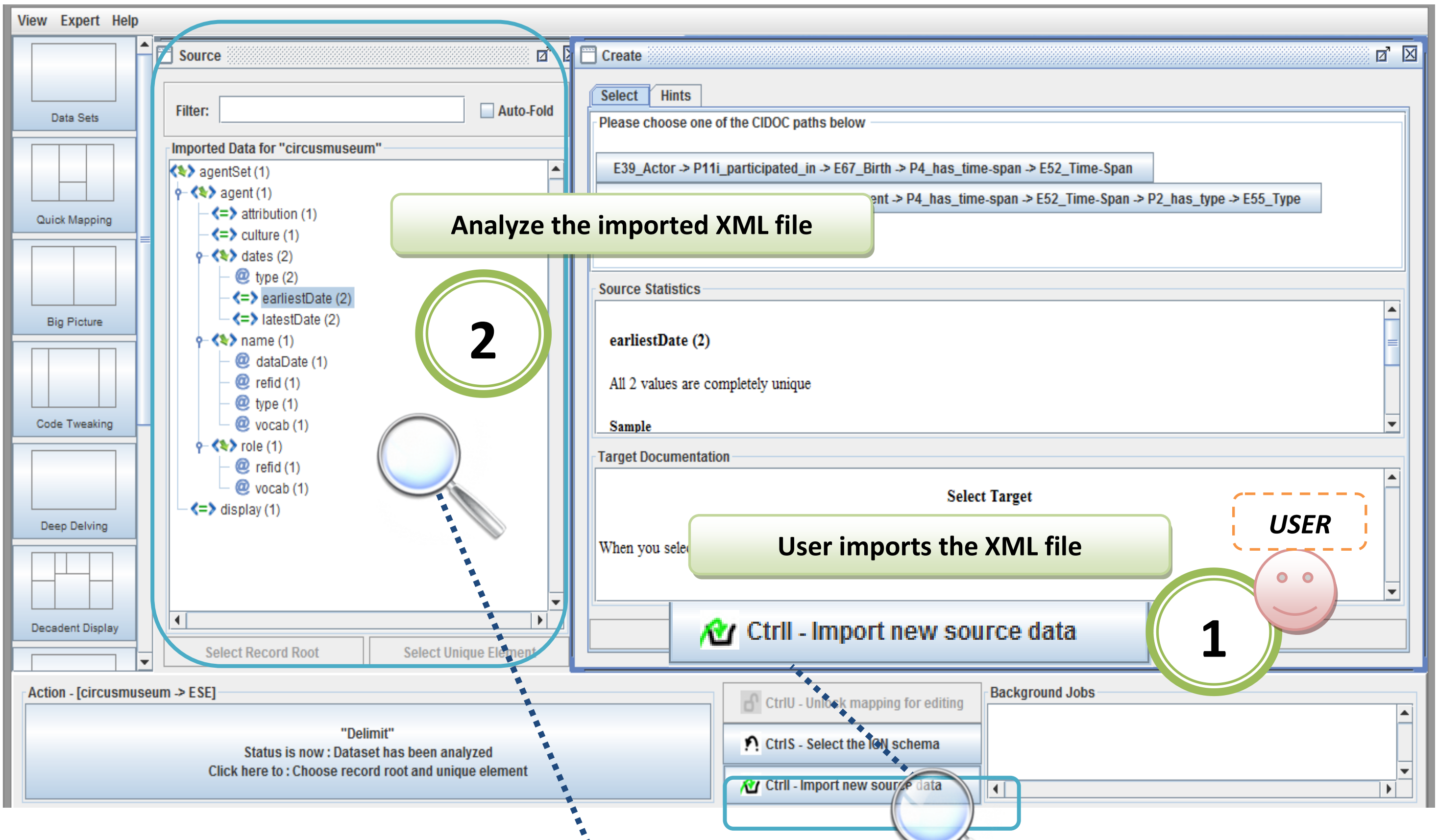
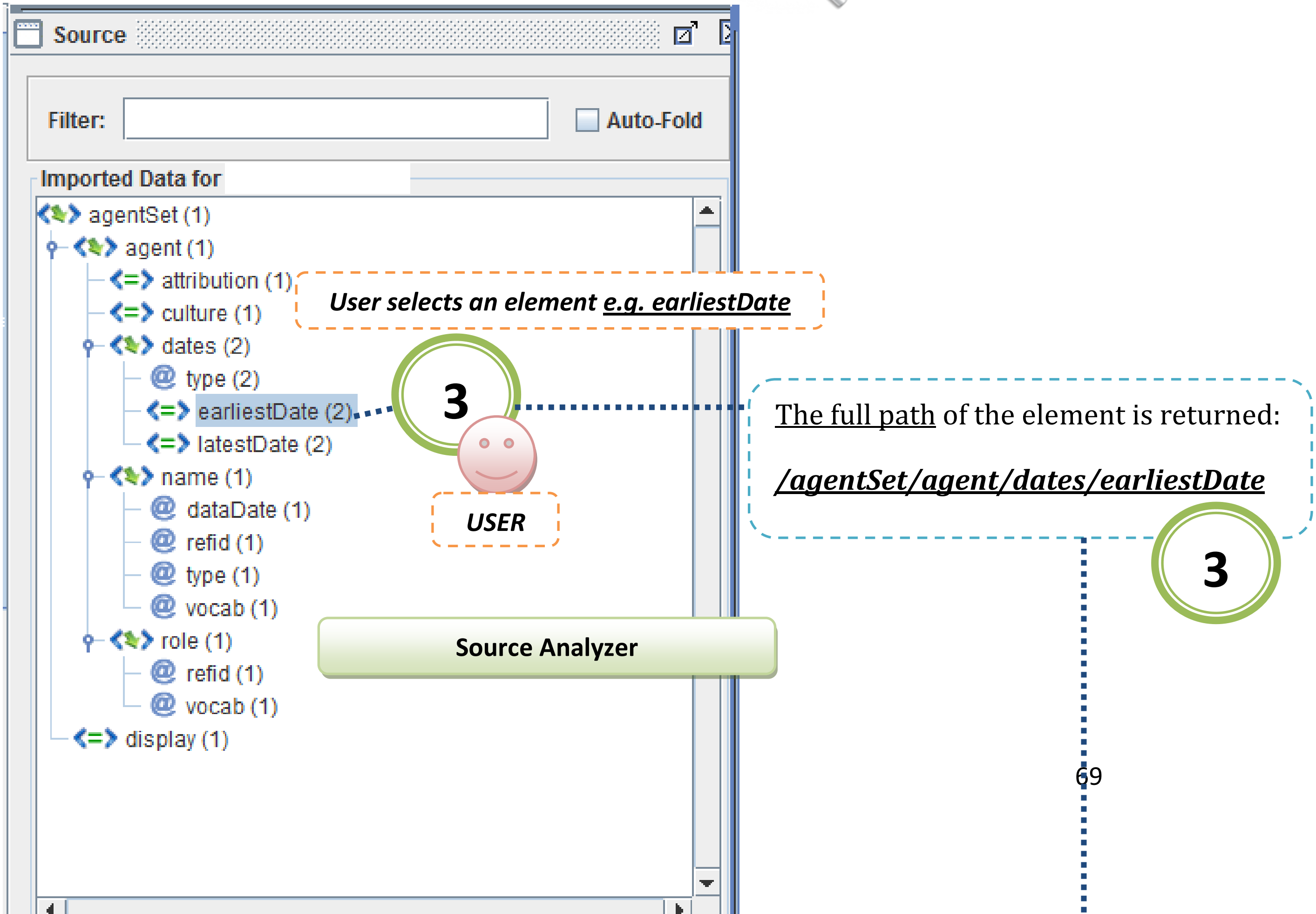

View Expert Help
Data Sets
Quick Mapping
Big Picture
Code Tweaking
Deep Delving
Decadent Display
Source
Filter:
□ Auto-Fold
Imported Data for "circusmuseum"
agentSet (1)
agent (1)
attribution (1)
culture (1)
dates (2)
type (2)
earliestDate (2)
latestDate (2)
name (1)
dataDate (1)
refid (1)
type (1)
vocab (1)
role (1)
refid (1)
vocab (1)
display (1)
Select Record Root
Select Unique Element
Create
Select
Hints
Please choose one of the CIDOC paths below
E39_Actor -> P11i_participated_in -> E67_Birth -> P4_has_time-span -> E52_Time-Span
ent -> P4_has_time-span -> E52_Time-Span -> P2_has_type -> E55_Type
Source Statistics
earliestDate (2)
All 2 values are completely unique
Sample
Target Documentation
Select Target
When you selec
Ctrl - Import new source data
Action - [circusmuseum -> ESE]
"Delimit"
Status is now : Dataset has been analyzed
Click here to : Choose record root and unique element
CtrlU - Unlock mapping for editing
CtrlS - Select the ION schema
Ctrl - Import new source data
Background Jobs
Analyze the imported XML file
2
User imports the XML file
USER
1

Figure 3.6 Delving SIP-Creator Source Analyzer View [2]



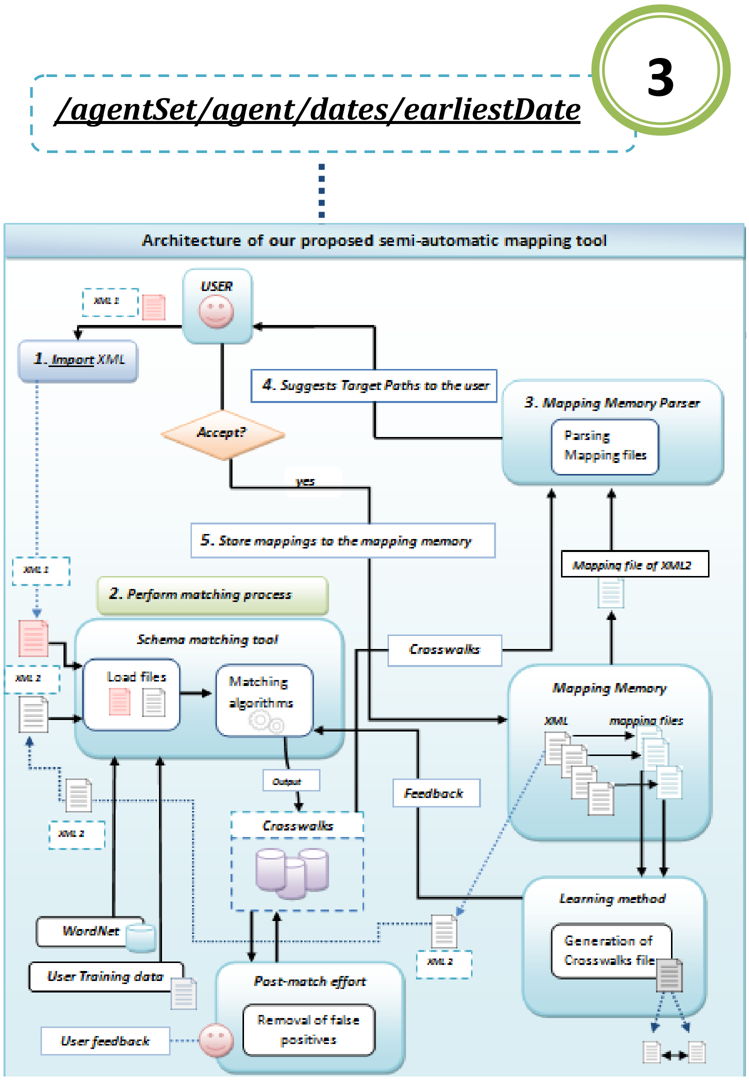
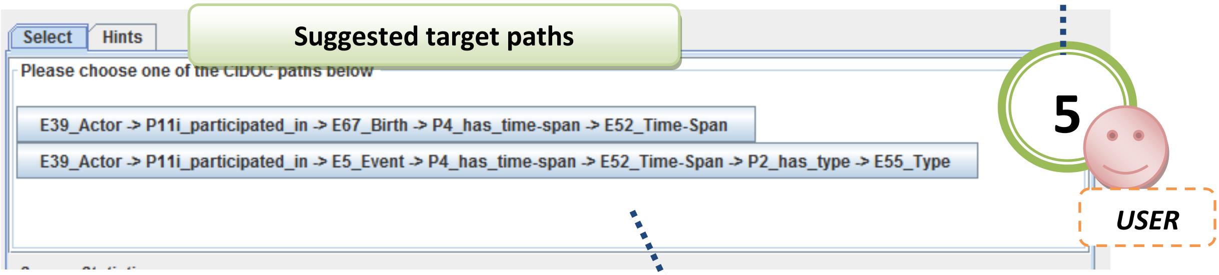


Figure 3.7 Architecture of our semi-automatic mapping tool

4

In Step 4 our system's modules are called (as we have already mentioned in section ***"Presentation of our proposed semi-automatic mapping tool")*** and the target paths are shown to the user in Step 5 So, the user can choose one of the target paths that are proposed to him

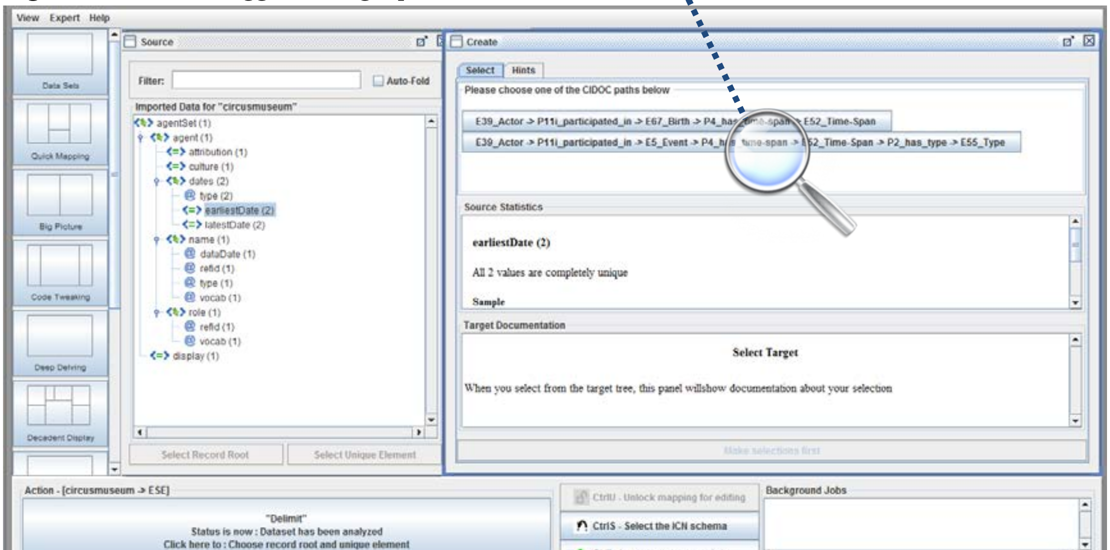


Please choose one of the CIDOC paths below

E39_Actor -> P11i_participated_in -> E67_Birth -> P4_has_time-span -> E52_Time-Span

E39_Actor -> P11i_participated_in -> E5_Event -> P4_has_time-span -> E52_Time-Span -> P2_has_type -> E55_Type

Figure 3.8 View of suggested target paths [1]



Imported Data for "circusmuseum"

```
agentSet (1)
  agent (1)
    attribution (1)
    culture (1)
    dates (2)
      type (2)
      earliestDate (2)
      latestDate (2)
    name (1)
      dataDate (1)
      refid (1)
      type (1)
      vocab (1)
    role (1)
      refid (1)
      vocab (1)
  display (1)
```

Please choose one of the CIDOC paths below

E39_Actor -> P11i_participated_in -> E67_Birth -> P4_has_time-span -> E52_Time-Span

E39_Actor -> P11i_participated_in -> E5_Event -> P4_has_time-span -> E52_Time-Span -> P2_has_type -> E55_Type

Source Statistics

earliestDate (2)

All 2 values are completely unique

Sample

Target Documentation

Select Target

When you select from the target tree, this panel willshow documentation about your selection

Action - [circusmuseum -> ESE]

"Delimit"
Status is now : Dataset has been analyzed
Click here to : Choose record root and unique element

Figure 3.9 View of suggested target paths [2]

When the user clicks on one of the suggested target paths, our CRM editor opens in a new Java frame. We analyze the CRM editor in the following section.

3.6.1 Target path editor (CRM Editor)

As we have already mentioned in the Introduction of this Chapter, we would like to **prevent** the user from doing the following steps in order to create a mapping file.

- Write manually the mapping files.
- Learn the XSD schema of the mapping description language. That means that he shall spend time studying it in order to fully understand it and in case he is not an IT expert, he is going to find difficulty in that.
- Form the target paths manually by checking the inheritance of classes and properties (no assistance module).
- Make the validation of all the target paths in order to ensure that the target paths confront to the rdfs schema of the target ontology.
- Check manually whether the XML mapping files are well-formed and don’t contain errors.

So, we created a Target path editor called “CRM editor” in order to simplify the mapping process and help the user map his data to the target ontology in an easy and friendly way, even if he is not an IT expert.

CRM Editor helps in creating target paths based on a specific target ontology. It provides specialized features for editing a target schema path. It displays the structure of the target schema path and includes the following features:

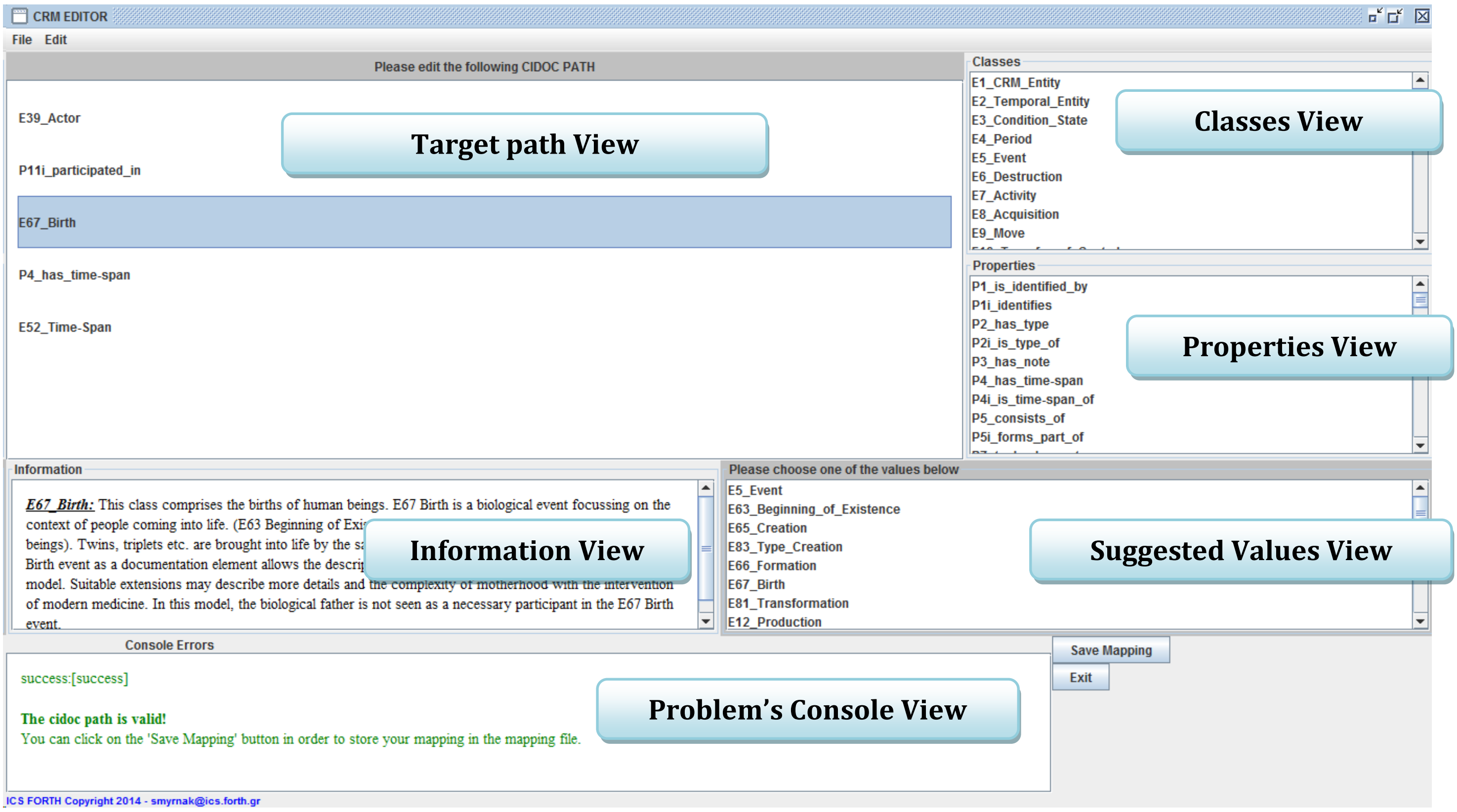
- Create a target schema path.
  The user can create long target paths (in seconds) without even writing a line of code.
- Edit the target schema path.
- Remove nodes from the path.
- Set internal nodes in the path.

- Set additional/constant nodes.
- Create and Save the mapping to the mapping file.
- Problems Console View
    - Error Highlighting.
    - Defining type of error.
    - Specifying position of the encountered error.
- Target path suggestion values module assistance
    - The user creates the paths really fast because this module suggests the right range values for every property and the right property values for every domain entity.
    - The target suggested module is analyzed in the Java class library.
- Target schema visualizer that is based on the target schema analyzer (which is a Java module)
    - Classes View.
    - Properties View.
    - Information Console View.
- Merging of target ontologies.
- Target Path Validator.

The Target Ontology Visualizer consists of the Classes View, the Properties View and the Information View. It depends on the Target Schema Analyzer Module.

I implemented the following Java application (by using java swing) in order to help the users edit the target schema path. The editor is a standalone Java frame that can be integrated in any Java application.

We can see a screenshot of the CRM editor.



E67 Birth: This class comprises the births of human beings. E67 Birth is a biological event focussing on the context of people coming into life. (E63 Beginning of Exis beings). Twins, triplets etc. are brought into life by the sa Birth event as a documentation element allows the descrip model. Suitable extensions may describe more details and the complexity of motherhood with the intervention of modern medicine. In this model, the biological father is not seen as a necessary participant in the E67 Birth event.

success:[success]

The cidoc path is valid!
You can click on the 'Save Mapping' button in order to store your mapping in the mapping file.

Figure 3.10 CRM editor – Graphical User interface - Main Layout

3.6.1.1 Menu View

This View describes the list of the items in the menu list.

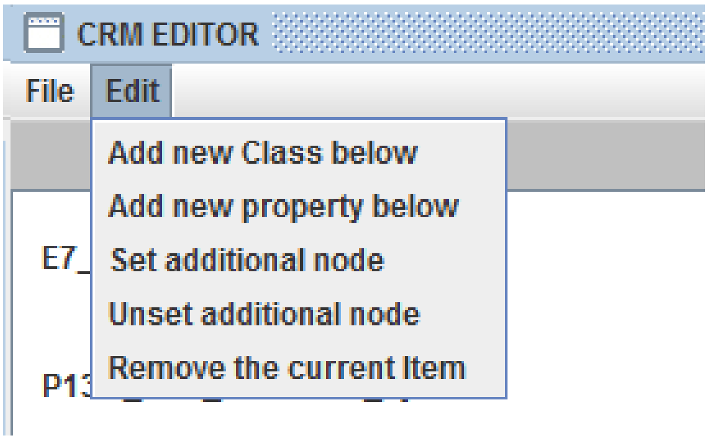
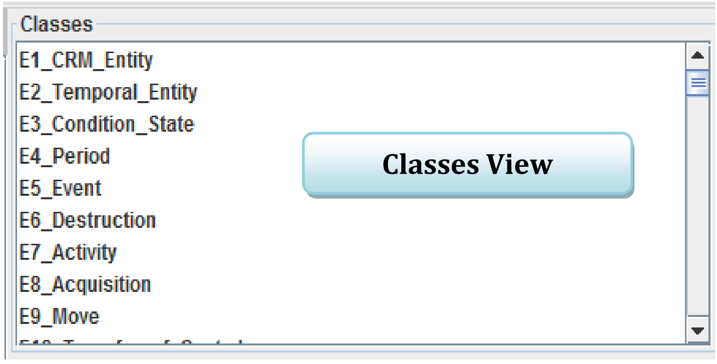


Figure 3.11 CRM editor - Menu

3.6.1.2 Classes view

The **Classes View** displays all the classes of the target ontology.

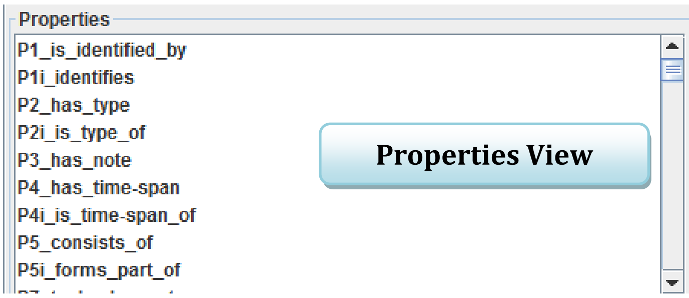


E1_CRM_Entity
E2_Temporal_Entity
E3_Condition_State
E4_Period
E5_Event
E6_Destruction
E7_Activity
E8_Acquisition
E9_Move

Figure 3.12 CRM editor - Classes View

3.6.1.3 Properties view

The Properties View lists all the properties of the target ontology.

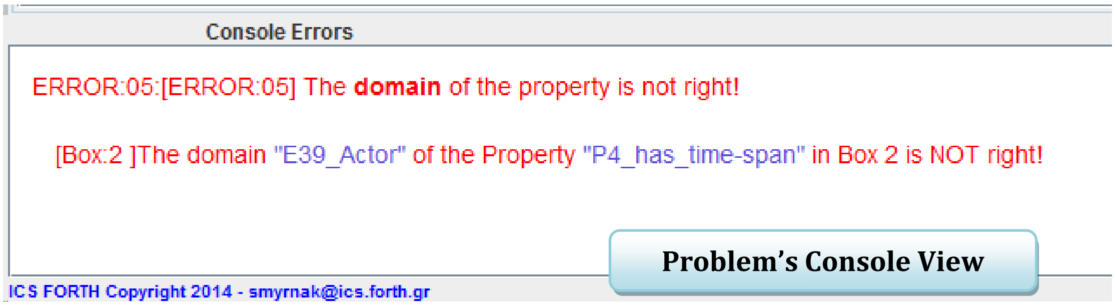


P1_is_identified_by
P1i_identifies
P2_has_type
P2i_is_type_of
P3_has_note
P4_has_time-span
P4i_is_time-span_of
P5_consists_of
P5i_forms_part_of

Figure 3.13 CRM editor - Properties View

3.6.1.4 Console view

The **Problems' Console View** displays the output of the validation process and highlights the errors encountered during the build process of the target schema path. This View lists the type of the error and the position of the error in the path. The validation module defined in Java Class Library is used here.



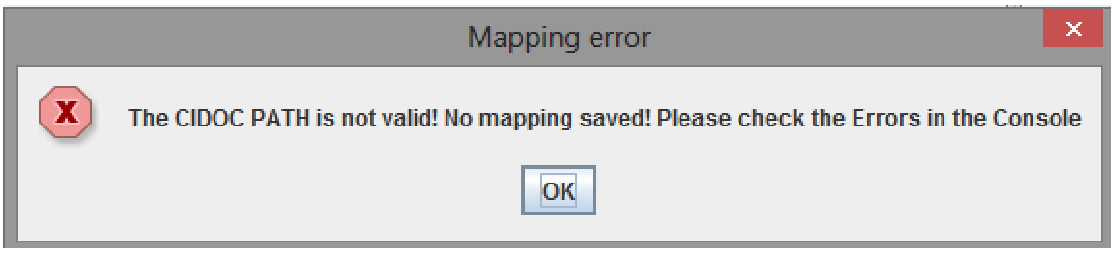
Console Errors

ERROR:05:[ERROR:05] The **domain** of the property is not right!

[Box:2]The domain "E39_Actor" of the Property "P4_has_time-span" in Box 2 is NOT right!

Figure 3.14 CRM-editor – Problems Console View (error info)

In the previous example the path was not valid. If the user tries to save the mapping to the mapping file, a box dialog appears indicating to the user that the path is valid.

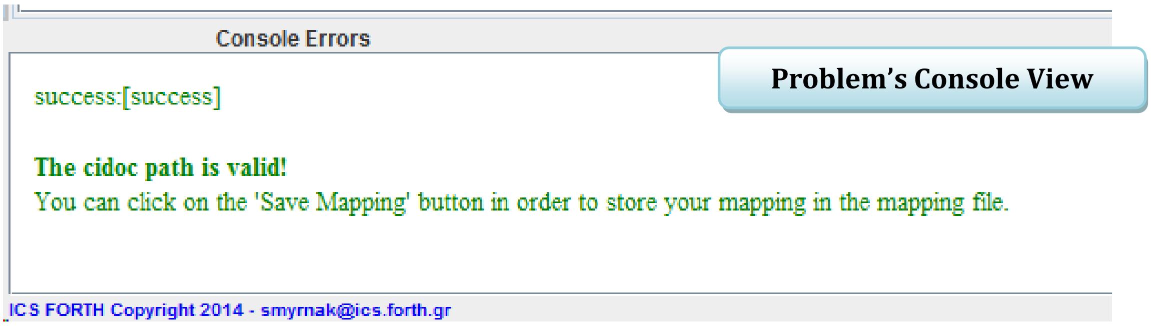


Mapping error

The CIDOC PATH is not valid! No mapping saved! Please check the Errors in the Console

Figure 3.15 CRM-editor - Mapping error dialog message

In case the path is valid, the text's color is in green and the mapping can be saved to the mapping file.



Console Errors

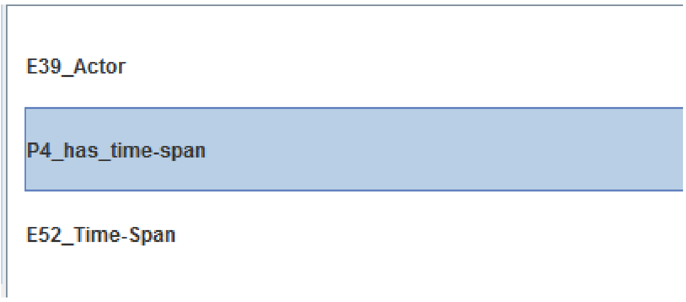
success:[success]

The cidoc path is valid!
You can click on the 'Save Mapping' button in order to store your mapping in the mapping file.

Figure 3.16 CRM editor – Problems Console View (success information)

3.6.1.5. Target path view

This View displays the target path.



E39_Actor

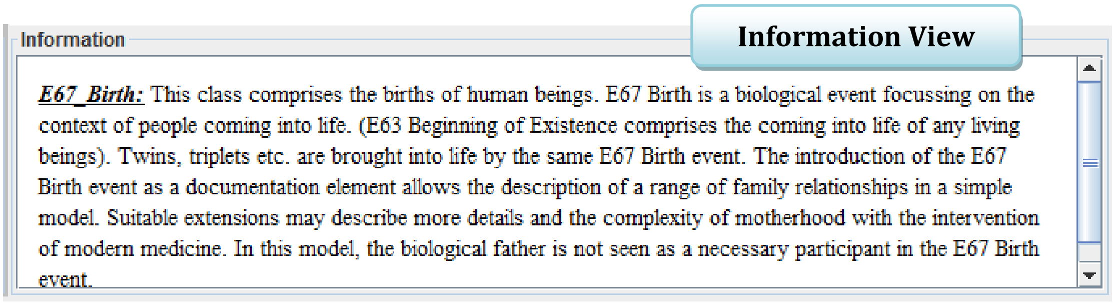
P4_has_time-span

E52_Time-Span

Figure 3.17 CRM editor – Target path View

3.6.1.6 Information's view

The **Information View** provides a description of the Class or Property. This documentation comes/ from the <rdfs:comment></rdfs:comment> in the resource description framework file (.rdfs) and describes the resource in a human readable text. “This textual comment helps clarifying the meaning of the RDF classes and properties”.

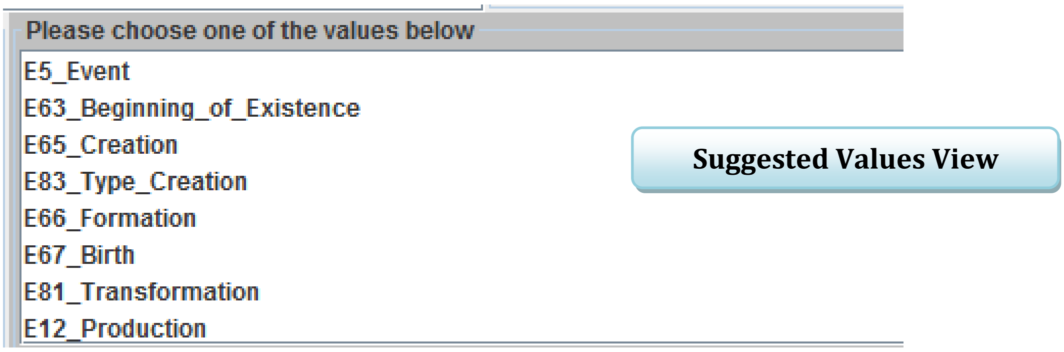


E67_Birth: This class comprises the births of human beings. E67 Birth is a biological event focussing on the context of people coming into life. (E63 Beginning of Existence comprises the coming into life of any living beings). Twins, triplets etc. are brought into life by the same E67 Birth event. The introduction of the E67 Birth event as a documentation element allows the description of a range of family relationships in a simple model. Suitable extensions may describe more details and the complexity of motherhood with the intervention of modern medicine. In this model, the biological father is not seen as a necessary participant in the E67 Birth event.

Figure 3.18 CRM-editor – Information View

3.6.1.7 Suggestion view

This View helps the user in order to form the path in the target ontology. While the user forms the path, the module suggests the right properties for every domain entity and the right ranges for every property, depending on the target ontology constraints which exist in the ontology.



Please choose one of the values below

E5_Event
E63_Beginning_of_Existence
E65_Creation
E83_Type_Creation
E66_Formation
E67_Birth
E81_Transformation
E12_Production

Figure 3.19 CRM editor – Suggested values View

3.7 Java Class library

3.7.1 Introduction

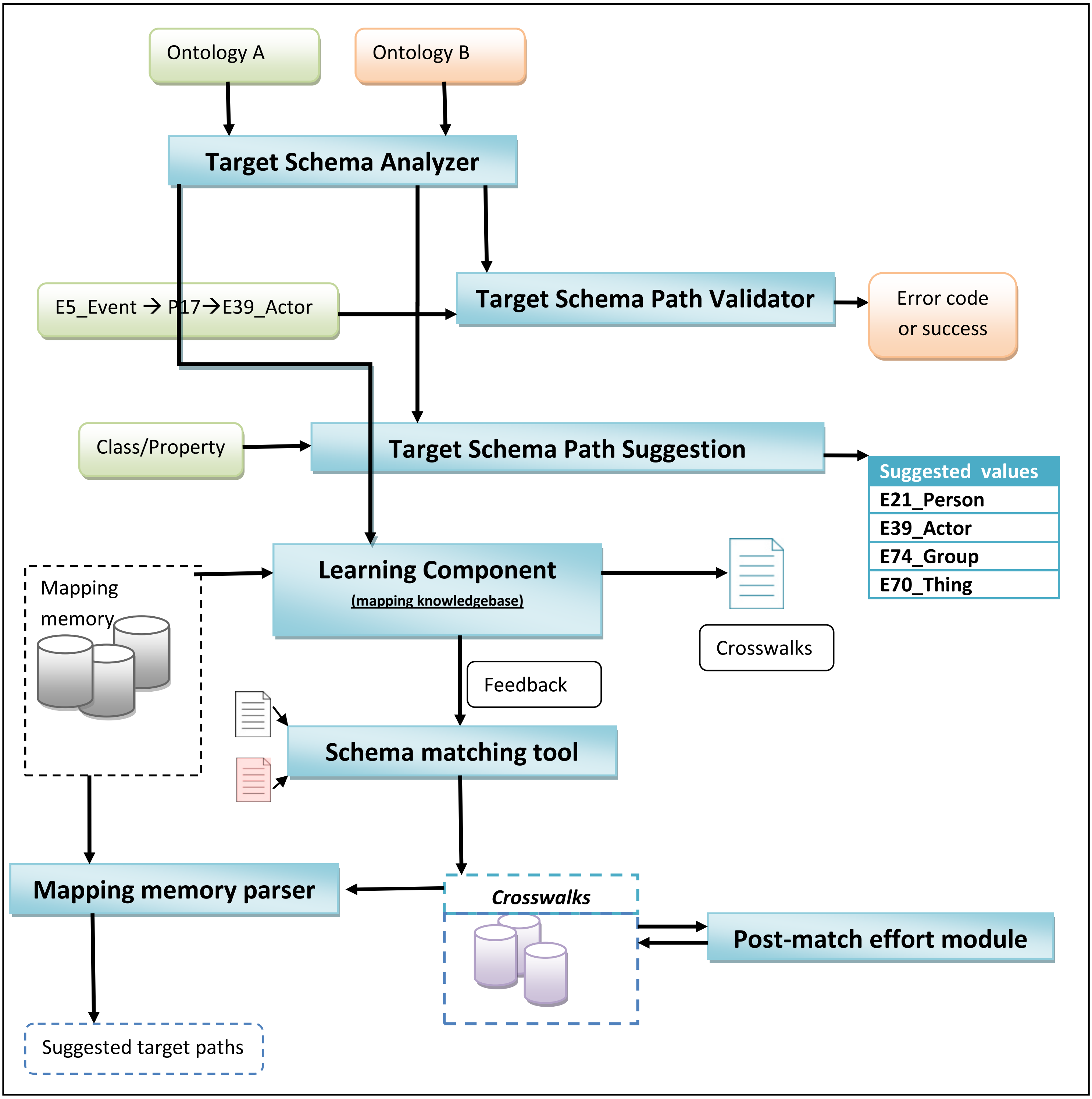


Figure 3.20 Java class library of our system

In this section, I describe the creation and implementation of a list of modules. I have created and implemented a **Java Class Library** that contains a set of dynamically loadable libraries which Java applications can call at run time. These dynamically loadable libraries are going to be used in the implementation of the CRM mapping tool.

The Java loadable libraries described in this section are:

- **Target schema analyzer module**
- **Target schema path validator module**
- **Target schema path suggestion module**
- **Learning method module**
- **Suggestion path module**
  - **Schema matching tool module**
  - **Mapping memory parser module**
- **Post-match effort module**

3.7.2 Target schema analyzer

The Target schema analyzer is a Java loadable library that has a variety of methods/functions that any Java application can call at run time.

The Target Schema Analyzer:

- Loads any kind of target schema ontology (RDFS) (e.g. cidoc-crm, frbr-crm etc)
  - User can import a list of resource description framework schema files (.rdfs files). These files are written based on the RDF schema http://www.w3.org/TR/rdf-schema/ .
- Merges extensions of ontologies (e.g. cidoc-crm, http://www.ics.forth.gr/isl/CRMdig/) into one target ontology.
- Parses and analyses the target schema.
- Stores all the information (classes/ subclasses/ superclasses/ properties/ information etc.) related to the target schema.
- Provides a variety of methods to the caller/programmer.

Figure 3.21 Target Schema Analyzer module architecture

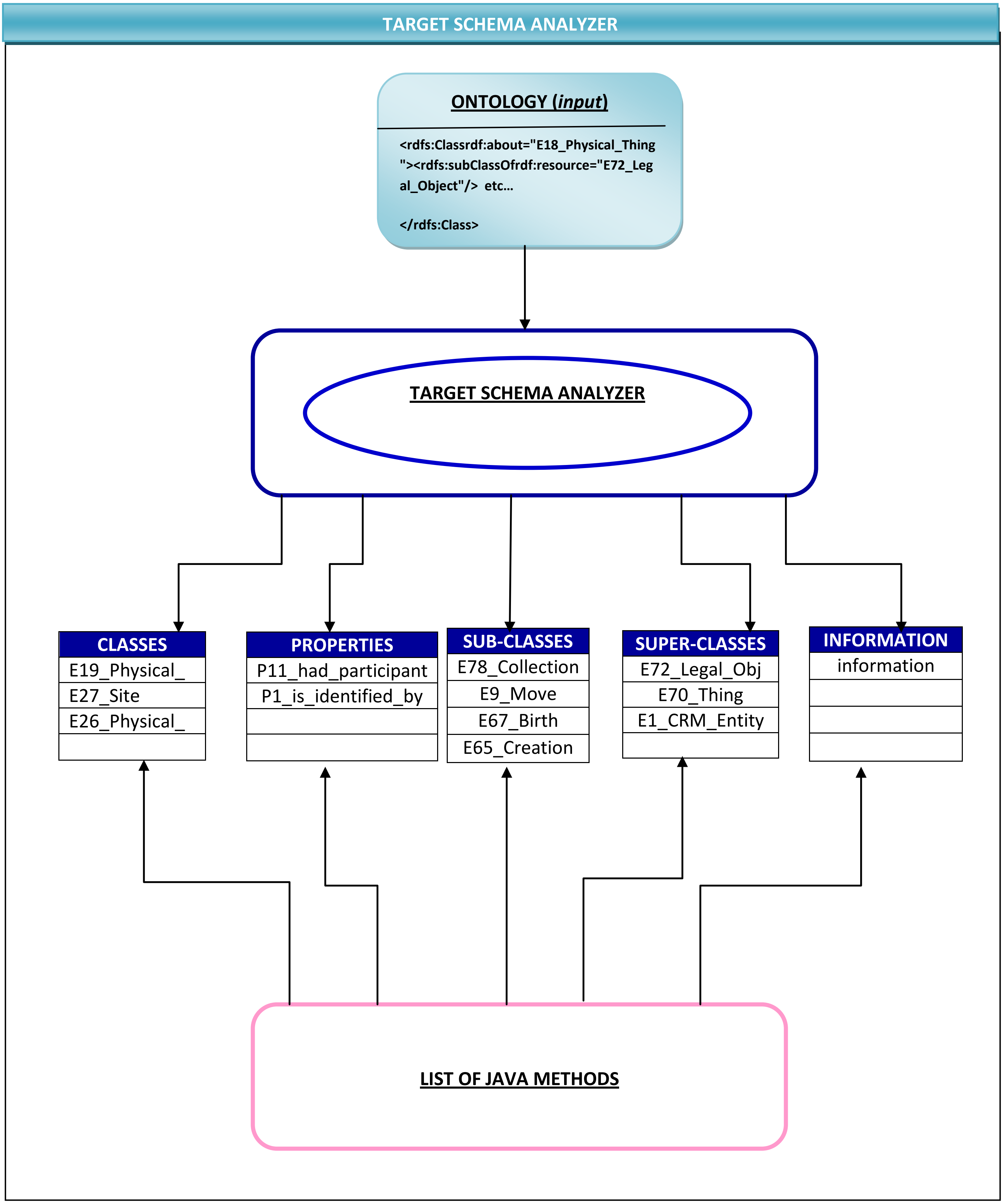
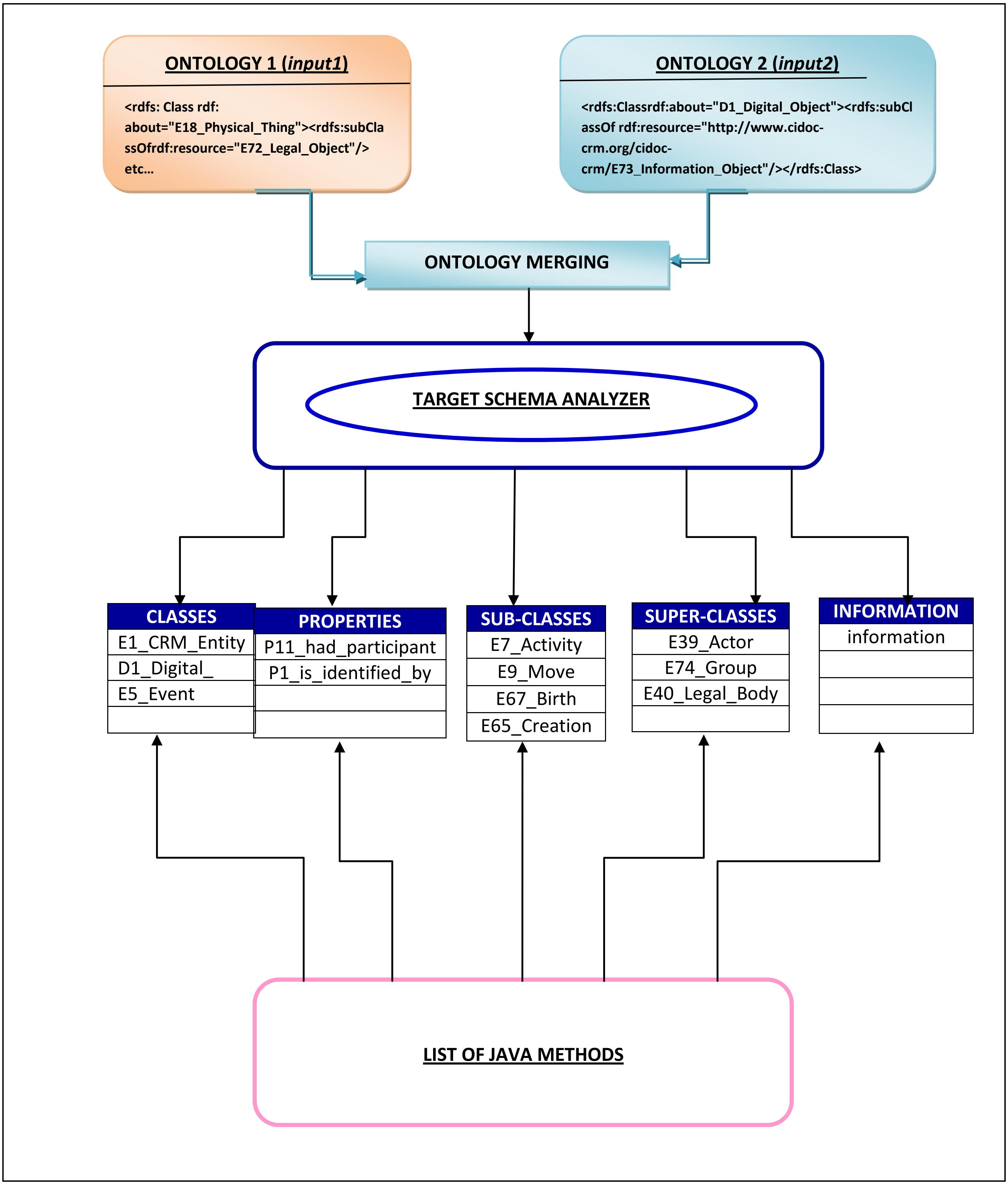


Figure 3.22 Target Schema Analyzer module - merging of target ontologies



As I mentioned before the Target Schema Analyzer analyzes and merges more than one target ontologies.

Example of an input (ontology) that is given to the Target Schema Analyzer is given below. This ontology is based on the RDFS schema[88] and it is a sample of the “Cidoc.rdfs” [89] file.

```
<rdf:RDFxml:lang="en" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.cidoc-crm.org/cidoc-crm/">


<rdfs:Classrdf:about="E1_CRM_Entity">
        <rdfs:labelxml:lang="el">Οντότητα CIDOC CRM</rdfs:label>
        <rdfs:labelxml:lang="en">CRM Entity</rdfs:label>
        <rdfs:labelxml:lang="de">CRM Entität</rdfs:label>
        <rdfs:labelxml:lang="ru">CRM Сущность</rdfs:label>
        <rdfs:labelxml:lang="fr">Entité CRM</rdfs:label>
        <rdfs:labelxml:lang="pt">Entidade CRM</rdfs:label>
        <rdfs:labelxml:lang="cn">CRM实体</rdfs:label>
        <rdfs:comment>This class comprises all things in the universe of discourse of
the CIDOC Conceptual Reference Model.
It is an abstract concept providing for three general properties:
1.      Identification by name or appellation, and in particular by a preferred identifier
2.      Classification by type, allowing further refinement of the specific subclass an
instance belongs to
3.      Attachment of free text for the expression of anything not captured by formal
properties
With the exception of E59 Primitive Value, all other classes within the CRM are directly
or indirectly specialisations of E1 CRM Entity.
</rdfs:comment>
</rdfs:Class>

<rdf:Propertyrdf:about="P1_is_identified_by">
        <rdfs:labelxml:lang="el">αναγνωρίζεται ως</rdfs:label>
<rdfs:labelxml:lang="de">wirdbezeichnetals</rdfs:label>
<rdfs:labelxml:lang="en">is identified by</rdfs:label>
<rdfs:labelxml:lang="ru">идентифицируетсяпосредством</rdfs:label>
<rdfs:labelxml:lang="fr">estidentifiée par</rdfs:label>
        <rdfs:labelxml:lang="pt">é identificadopor</rdfs:label>
        <rdfs:labelxml:lang="cn">有识别称号</rdfs:label>
        <rdfs:comment>This property describes the naming or identification of any real
world item by a name or any other identifier.
This property is intended for identifiers in general use, which form part of the world the
model intends to describe, and not merely for internal database identifiers which are
specific to a technical system, unless these latter also have a more general use outside
the technical context. This property includes in particular identification by mathematical
expressions such as coordinate systems used for the identification of instances of E53
```

[88] http://www.w3.org/TR/rdf-schema/

[89] http://www.cidoc-crm.org/rdfs/cidoc_crm_v5.1-draft-2013May.rdfs

```
Place. The property does not reveal anything about when, where and by whom this
identifier was used. A more detailed representation can be made using the fully
developed (i.e. indirect) path through E15 Identifier Assignment.
</rdfs:comment>
        <rdfs:domainrdf:resource="E1_CRM_Entity"/>
        <rdfs:rangerdf:resource="E41_Appellation"/>
</rdf:Property>

<rdf:Propertyrdf:about="P1i_identifies">
        <rdfs:labelxml:lang="de">bezeichnet</rdfs:label>
        <rdfs:labelxml:lang="ru">идентифицирует</rdfs:label>
        <rdfs:labelxml:lang="fr">identifie</rdfs:label>
<rdfs:labelxml:lang="el">είναι αναγνωριστικό</rdfs:label>
<rdfs:labelxml:lang="en">identifies</rdfs:label>
        <rdfs:labelxml:lang="pt">identifica</rdfs:label>
        <rdfs:labelxml:lang="cn">被用来识别</rdfs:label>
        <rdfs:domainrdf:resource="E41_Appellation"/>
        <rdfs:rangerdf:resource="E1_CRM_Entity"/>
</rdf:Property>
</rdf:RDF>
```

Table 3.7 CIDOC-CRM ***cidoc.rdfs*** file

The methods that are provided in this module are depicted below: The Javadoc files describe in detail the type of inputs, outputs and contain additional information for the methods.

Methods and Description	***Returns***
getAllClasses**()**	Returns a List containing all the classes of the Target Schema
getAllClassesInformation**()**	Returns a Collection containing all the information of all the classes in the rdf target schema
getAllProperties**()**	Returns an List containing all the properties of the Target Schema
getAllPropertiesInformation**()**	Returns a Collection containing all the information of all the properties in the rdf target schema
getClassInformation**(className)**	Returns a String containing the **textual comment** for the rdf Class. Information comes from the *rdfs:comment* tag in *.rdfs* file The *rdfs:comment* provides a human-readable description of a

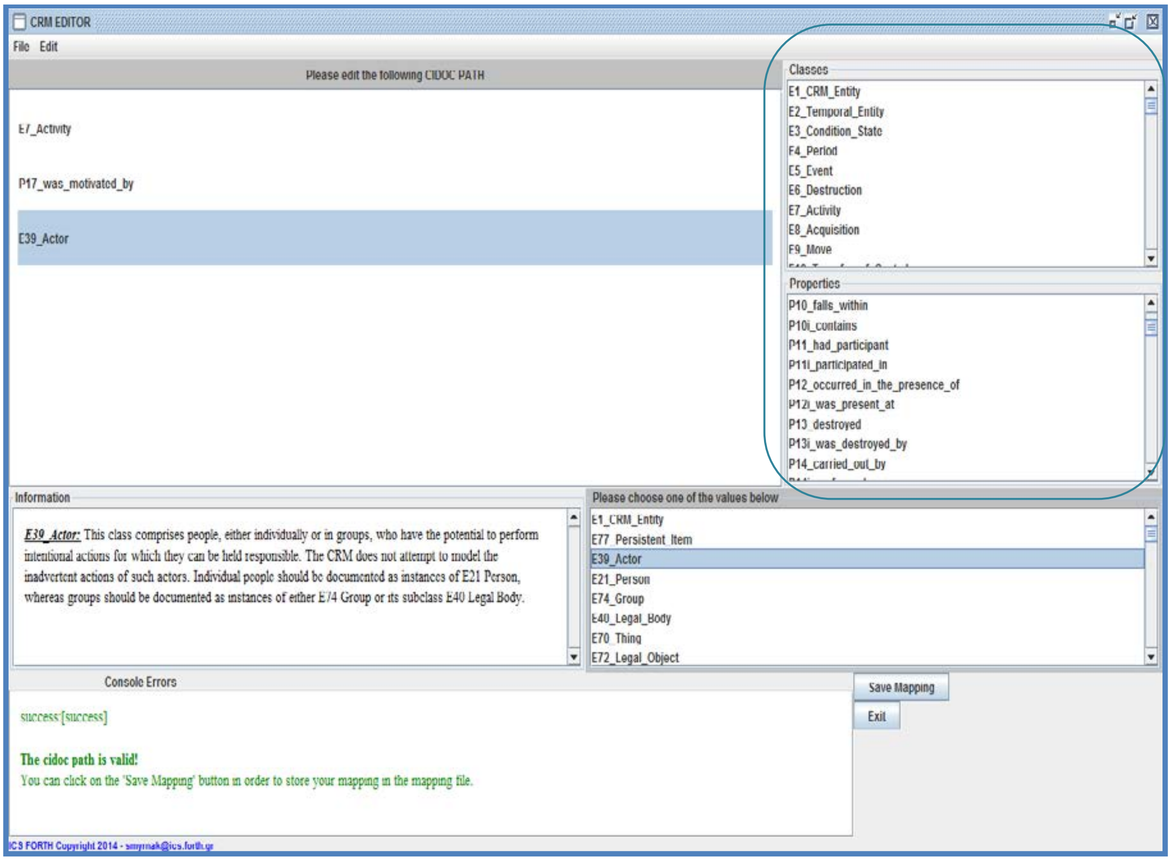
	resource.
getDomainOfProperty**(Property1)**	Returns the Domain of a Specific property in the Target Schema
getPropertyInformation**(propertyName)**	Returns a String containing the **textual comment** for the rdf Property. Information comes from the *rdfs:comment* tag in *.rdfs* file The *rdfs:comment* provides a human-readable description of a resource.
getRangeOfProperty**(Property1)**	Returns the Range of a Specific property in the Target Schema
getSubclassesOfClass**(class1, SubClasses)**	returns all the subclasses of a specific class
getSuperClassesOfClass**(Class1)**	returns all the **superclasses** of the Class
isClass**(str)**	Returns **true** if str is a Class and **false** if str is not a Class
isProperty**(str)**	Returns **true** if str is a Property and **false** if str is not a Property
IsSubClassOfClass**(Class1, java.lang. Class2)**	**true** if Class1 is subclass of Class2 and **false** if Class1 is not a subclass of Class2
IsSubPropertyOf**(Property1, Property2)**	**true** if Property1 is subproperty of Property2 and **false** if Property1 is not a subproperty of Property2
ReturnAllClassesHashMap**()**	Returns a HashMap storing the immediate **SuperClass** of every Class in the rdf target Schema
ReturnAllPropertiesHashMap**()**	Returns a HashMap storing the immediate **SuperProperty** of every Property in the rdf target Schema

Table 3.8 Description of methods of the target schema analyzer

Java API Documentation and Javadoc files are available for the Target Schema Analyzer.

The Java API is entailed in order to provide a list of important functions to the programmer in order to use them in the construction of the GUI that is going to describe the target ontology (target Schema Visualizer). The GUI will help the user understand the target ontology by using all the information included in the rdfs file. Also, most of these methods are called by the other modules that are described later (the Target Schema Path Validator [checks if a target path is valid], the Target Schema Value Suggestion module, Learning method module).

For instance, I call the methods of this module in order to create the Graphical User Interface of the CRM editor, as depicted below.



E39_Actor: This class comprises people, either individually or in groups, who have the potential to perform intentional actions for which they can be held responsible. The CRM does not attempt to model the inadvertent actions of such actors. Individual people should be documented as instances of E21 Person, whereas groups should be documented as instances of either E74 Group or its subclass E40 Legal Body.

success:[success]

The cidoc path is valid!

You can click on the 'Save Mapping' button in order to store your mapping in the mapping file.

ICS FORTH Copyright 2014 - smyrnak@ics.forth.gr

Figure 3.23 CRM editor – Graphical User Interface

3.7.3 Target schema path validator

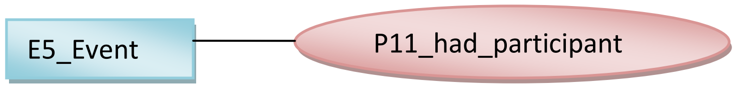
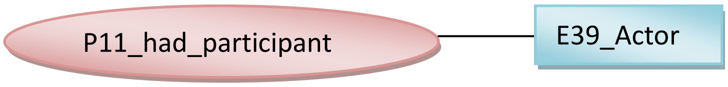
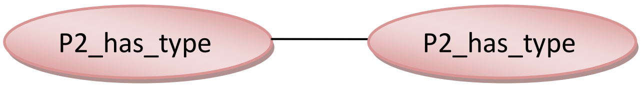
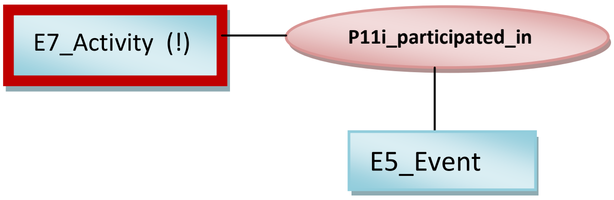
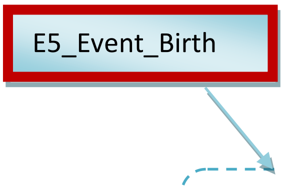
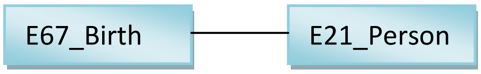
The Target schema path validator

- takes as input a path in the target ontology
- checks if the path confronts to the schema of the target ontology and it is valid
- returns the type of error or success, if the path is valid
    - returns also the position of the error in the target path

This Java Library uses the methods of the "Target Schema Analyzer" Java library in order to check if a path is valid. A JAVA API and Javadoc file are included in that module too.

I have created the following table describing the types of error codes in the **Target Schema Path Validator**.

TYPES OF ERRORS	ERROR DESCRIPTION
ERROR:01	The target path is empty.
ERROR:02	The target path is not complete, it contains only two nodes.
ERROR:03	The target path starts with a property and not a Class.
ERROR:04	Two Properties in a row.
ERROR:05	The domain of a specific property is not valid.
ERROR:06	A specific node in the target path is not a class or property. Does not exist in RDFS schema of the target ontology.
ERROR:07	Two Classes in a row.
ERROR:08	The range of a specific property is not valid.
ERROR:11	The target path shall end with a Class and not a Property.

TYPES OF ERRORS	Graphical Representation of invalid target paths
ERROR:02 The target path is not complete.	
ERROR:03 The target path shall start with a class and not a property.	
ERROR:04 Two properties in a row.	
ERROR:05 The **domain** of the property "P11i_participated_in" is not right/valid!	
ERROR:06 A specific node is not a class or a property. Not included in the resource description framework file of the target schema ontology.	 *The "**E5_Event_Birth**" class is not included in the RDFS schema of the target ontology (cidoc-crm).*
ERROR:07 Two classes in a row. A property is missing in the middle.	

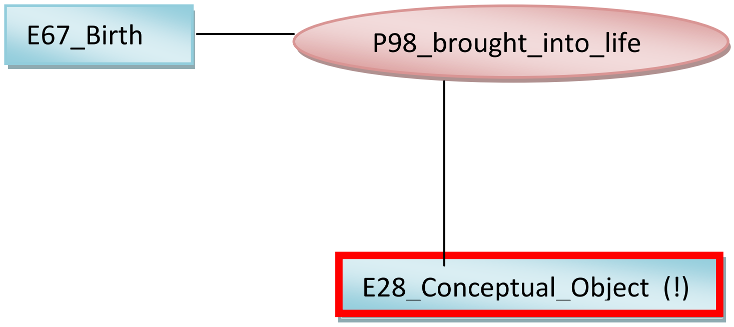
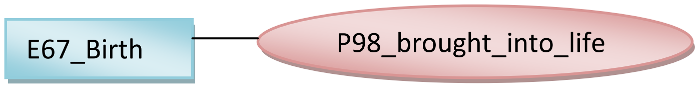
ERROR:08	 E67_Birth P98_brought_into_life E28_Conceptual_Object (!)
The **range** of the property “P98_brought_into_life” is not valid. The right range would be “E21_Person”.	
ERROR:11	
The target path ends with a property and not a class.	

Table 3.9 Target schema path validator – types of error codes

Methods and Description
ConstraintControl(**CrmLoader** crmL, java.util.ArrayList<java.lang.String> TARGET_PATH) This method returns a String that describes the type of error: ERROR:01 The target path is empty ERROR:02 The target path is not complete, it contains only two nodes ERROR:03 The target path starts with a property and not a Class. ERROR:04 Two Properties in a row. ERROR:05 The domain of a specific property is not valid. ERROR:06 A specific node in the target path is not a class or property.
PrintAdditionalError(java.lang.String str) Returns a String that analyses in detail in which position of the target path the error is located.
ReturnTextError(java.lang.String code) Returns a String describing the Type of Error

Table 3.10 Description of methods of the target schema path validator

3.7.4 Target schema path suggestion module

Since the user may not remember the target ontology, the inheritance of properties from superclasses to subclasses and all the other information by heart, I created an additional module in order to suggest the right values to the user.

This module helps the user in order to form the path in the target ontology. While the user forms the path, the module suggests the right properties for every domain entity and the right ranges for every property, depending on the target ontology constraints which exist in the ontology.

e.g. If we have a class and we would like to add a property to it, the module automatically searches for the properties of the superclasses of that class in order to suggest the right properties for that domain entity. An example is included below:

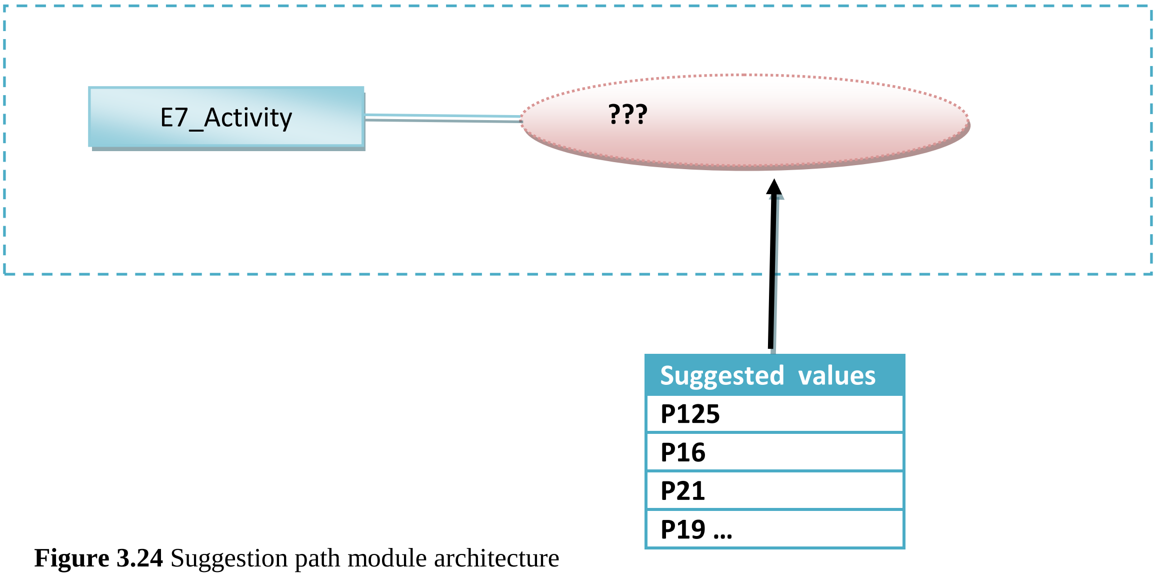


Figure 3.24 Suggestion path module architecture

3.7.5 Learning component module

The Learning component module acts as a schema matching tool, produces crosswalks and creates the training set for the learning method.

- Loads two mapping files from the mapping memory.
- Executes the algorithm described in the section “Learning method”.
- Stores the crosswalks in an XML file. This file is an xml instance document that refers to the XSD schema “CrosswalksMappingLanguage.xsd”.

- Gives feedback to the schema matching tool in order to improve its accuracy.

3.7.6 Suggestion path module

This module consists of two modules:

1. Schema matching tool module
2. Mapping memory parser

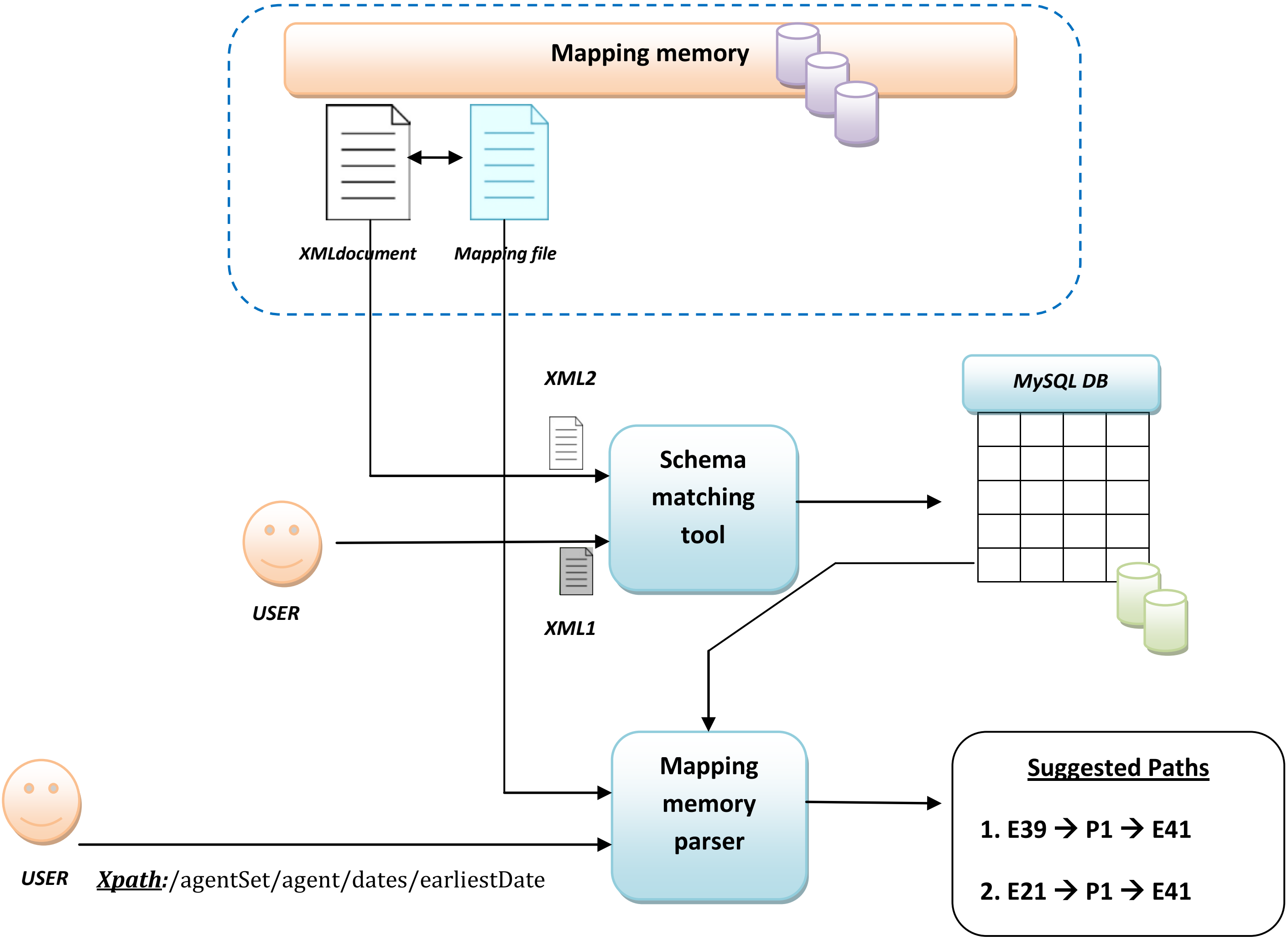


Figure 3.25 System architecture

3.7.6.1 Schema matching tool module

We have used and expanded the schema matching tool developed in the previous master theses[90,91,92]. The expanded schema matching tool provides a set of match algorithms and combines linguistic and structural matching methods. The inputs of the tool are XML document files and the output is a set of correspondences between those files.

This Java module:

- Loads two XML document files
    - The first XML document is the user's XML document that he would like to map to the target ontology
    - The second XML document comes from the Mapping memory. As we already know the mapping memory consists of pairs of XML-Mapping files
- Stores the matches between the two XML documents in the MySQL database.

3.7.6.2 Mapping memory parser

The mapping memory parser module:

- Takes as inputs, the xpath of the selected element
- Parses the mappings files based on the crosswalks output by the matching tool (we make use of a Java DOM parser)
- Outputs a list of suggested paths.

[90] Manakanatas, D., "Design and Implementation of a tool for semi-automated semantic schema matching", Master of Science Thesis Computer Science Department, University of Crete", E-locus University of Crete Institutional Repository, February 2006. Available from: http://elocus.lib.uoc.gr/dlib/c/5/7/metadata-dlib-2006manakanatas.tkl

[91] Kalaitzaki, M., : "Design and implementation of a system for semantic schema matching. Master of Science Thesis Computer Science Department, University of Crete", 2009. Available from: http://elocus.lib.uoc.gr/dlib/6/5/5/metadata-dlib-4d555a739dd11f5df7c7801d0be78ef8_1275560441.tkl

[92] Daskalaki, E., "Development and experimental evaluation of an ontology to ontology schema & instance matching system, Master of Science Thesis Computer Science Department, University of Crete", E-locus University of Crete Institutional Repository, 2011, Available from: http://elocus.lib.uoc.gr/dlib/d/d/b/metadata-dlib-1322814460-437568-20114.tkl

3.7.7 Post match effort module:

In this module the user accepts or denies one of the matches found by the schema matching process and he removes the false positives from the results.

Chapter 4

4 Evaluations

4.1 Dataset and metrics

In our system we have used a dataset that consists of several XML document files encoded in different metadata standards such as (LIDO,VRA,MODS, CDWA Lite, Dublin Core). These metadata standards are widely used and belong to different domains (e.g. museum domain, archives domain, library domain). Even if the metadata standards come from the same domain they vary in semantics and structure. Also, we have used in-house schemas.

For the evaluation we have used match quality measures such as Precision, Recall and F-measure [93].

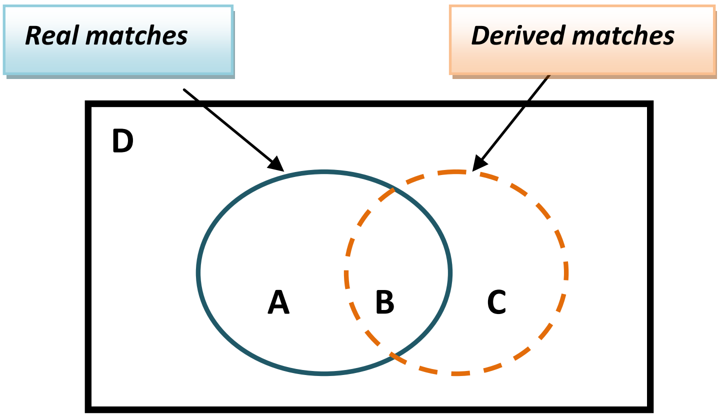


Figure 4.1 Real and derived matches

A: False negatives are matches needed but not automatically found by the schema matching process.

B: True Positives

C: False positives are matches falsely found by the system.

D: True negatives are false matches which correctly are not found by the system.

Table 4.1 Definition of false/true negatives/positives.

[93] Do, H.H., Melnik, S., Rahm, E., "Comparison of Schema Matching Evaluations", Web, Web-Services, and Database Systems, vol. 2593, pp. 221 – 237. Springer, 2003.

Metrics:

- **Precision =** | B | / | B| + | C|

- **Recall =** | B | / | A | + | B|

- **F-measure:** 2 * ((P * R) / (P + R))

In order to evaluate the quality of the schema matching results we found the correspondences/crosswalks between different metadata schemas manually. These manually matching results are used as the gold standard for the evaluation of the schema matching process. As we have already mentioned in previous chapters, we use a schema matching tool in order to suggest mappings to the user. When the right matches are found by the schema matching tool, the right target paths of the specified ontology (e.g. CIDOC) are shown to the user.

4.2 Tuning process

During the tuning process, I evaluated the schema matching tool in order to check its accuracy.[94] Tuning a matching tool increases the matching quality and reduces the post-match effort (the effort to remove the false positives from the final matching results and add the false negatives to them). Tuning is performed in the pre-match phase before the schema matching process begins. In this phase the human expert sets the rights parameters in the schema matching process (e.g. threshold, user training data, matcher algorithm etc) in order to improve the matching quality. For example, [95]a threshold that is set too low introduces false positives while a threshold that is too high may introduce false negatives.

As we have already mentioned in previous chapters, we have implemented and used a Learning method in order to create a training set for the schema matching tool. We re-use the mapping files that are already stored in the mapping memory in order to

[94] Bellahsene, Z., Duchateau, F., “Tuning for Schema Matching”, Schema Matching and Mapping , pp. 293-316. Springer, 2011.

[95] Gal, A., “Enhancing the Capabilities of Attribute Correspondences”, Schema Matching and Mapping, pp. 53-73. Springer, 2011.

produce a training set. This training set is a knowledge base that is used by the matchers and improves the Precision and Recall of the system. Of course, the quality of the training data that come from already mapped files affects the results.

Through my observation I figured out that the Learning method and the user training data improved the Recall of the system.

1. Question: Did the following schema matching strategies improve the Recall of the system?

1. **Learning method**
2. **User Training data**

Answer:

Yes, the Recall was improved.

Recall = B / (A + B) = (true positives) / (false negatives) + (true positives) =
= (matches that the system found) / Real matches.
(A + B) = Real matches

We noticed that the true positives increased and the false negatives were reduced.
For example:
If (A+B) (=Real matches) = 10 (the matches that the system shall find)

- If B = 5, R = 5/10 = 0.5
- If B =6, R = 6/10 = 0.6
- If B = 10, R = 10/10 = 1 , the best score 1 here.

When B (Real matches) in Recall formula increases, the numerator increases and the Recall is improved.

2. Question: When did you get a low Precision and why?

Answer:

The precision was affected by the false positive matches. As the false positive results increased, the Precision was getting low.

In our experiments, we observed that the schema matching tool found false positives matches when it was matching name tags.

Xpath1: /actor/name

Xpath2: /place/name

3.Question: So, what did you do in order to decrease the number of false positives matches and improve the Precision of the schema matching process?

Answer:

In that case we improved the structure matcher mechanism in order to avoid the false positive matches in our results. That matcher took into consideration the internal structure of the XML document and checked for the parents of each one of the node elements in the XML document.

4.3 Results

We evaluated the effectiveness of our proposed system by mapping a dataset of XML document files encoded in different metadata standards e.g. CIDOC, CDWA,MODS, VRA,DUBLIN CORE to the CIDOC-CRM target ontology.

We performed the mapping process twice:

1st run: We started mapping the dataset manually without using the semi-automatic tool that we have proposed in this thesis.

2nd run: We mapped the dataset with the assistance of our proposed semi-automatic mapping tool:

The table below describes the evaluation:

	Time (minutes)
Manual mapping *(1st run)*	**98**
Mapping with our proposed semi-automatic mapping system *(2nd run)*	**45**

Table 4.2 Comparison between manual and semi-automatic mappings

In the **manual mapping** the user shall:

1) analyze the source schema in order to define the full path of every element in the XML document tree,

2) form the target paths of the target ontology on his own, without the assistance of our suggestion module in our semi-automatic tool,

3) find the inheritance of classes and properties by analyzing the target schema in order to define the superclasses, subclasses, sub properties etc

4) study the mapping description language in order to form the mappings file,

5) write the mapping file manually,

6) check that the names of the classes/properties are written correctly based on the target schema ontology,

7) check if the mapping file is well-formed, it doesn't contain any error and it is formed based on the XSD schema of the mapping description language,

8) validate the target paths according to the rdfs schema of the target ontology.

All these steps shall be performed by one actor. In case the user finds difficulty in forming the mappings file another actor is going to get involved in this process in order to create the mappings.

The second run shows that the time goes down sharply when our proposed system is used because:

1) The system provides a friendly Graphical User Interface (GUI) with a source and a target analyzer (no need to for the user to browse the ontology in order to find the classes and properties)

2) The CRM editor creates the mapping files automatically.

3) Suggestion modules help the mapping process by re-using the mappings stored in the mapping memory.

4) Also, the target schema path suggestion module suggests the right properties for every domain entity and the right ranges for every property, depending on the target ontology constraints which exist in the ontology.

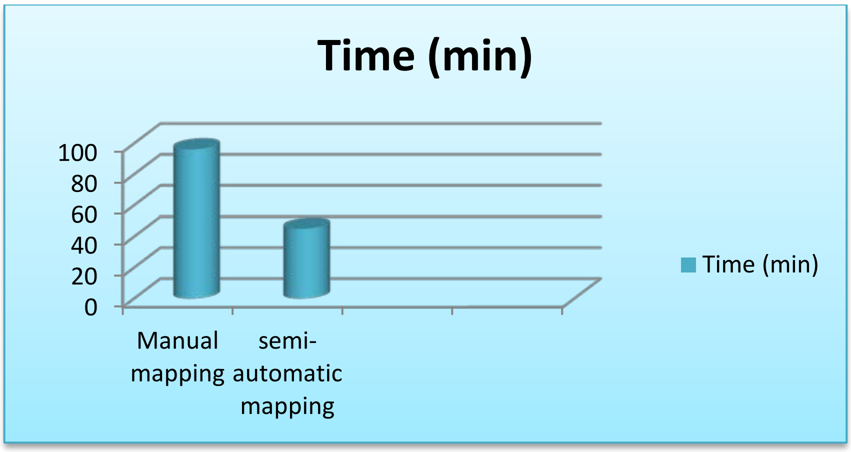


Figure 4.2 Time in minutes for manual and semi-automatic mapping

4.3.1 Mapping memory experiment

As the amount of mapping files was growing in the mapping memory we noticed that the Recall was improved. As the mapping knowledge base was enriched by many mapping files the size of the training set increased and true positives matches were added to the result .

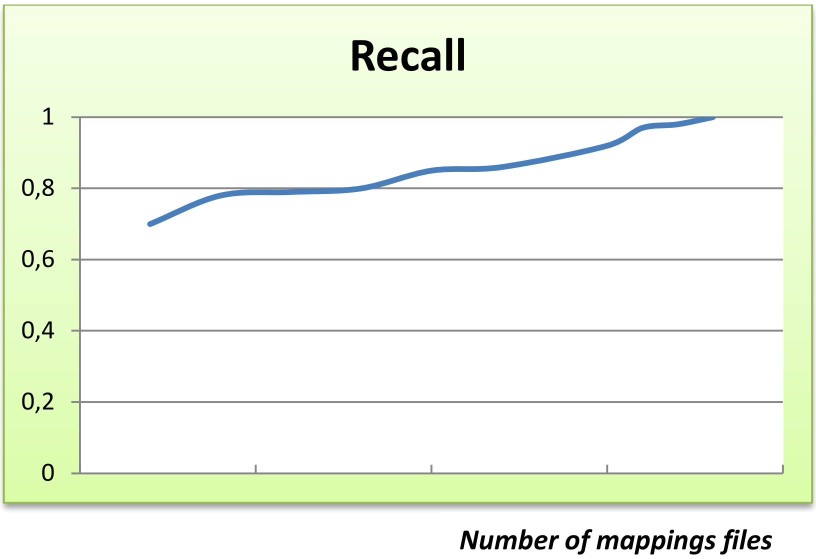
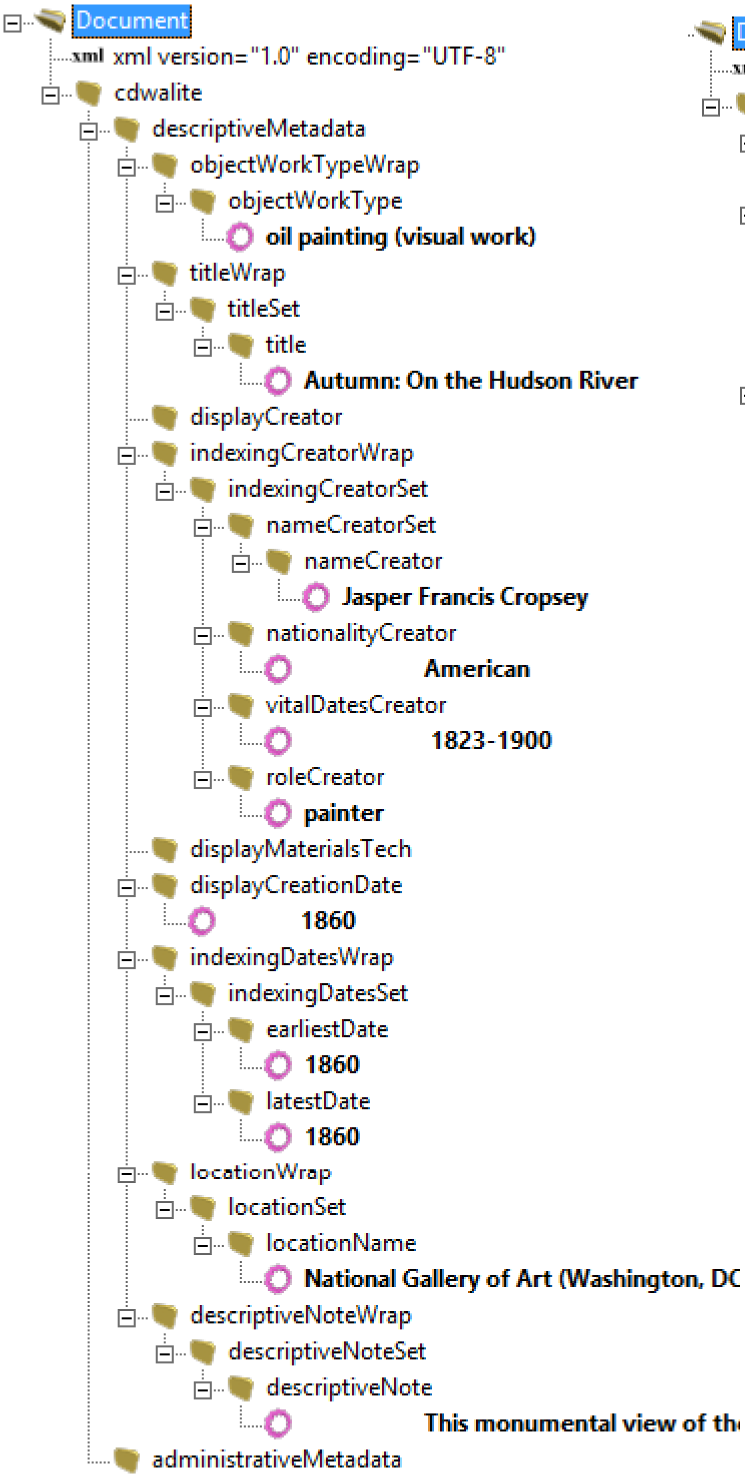


Figure 4.3 Recall affected by number of mapping files

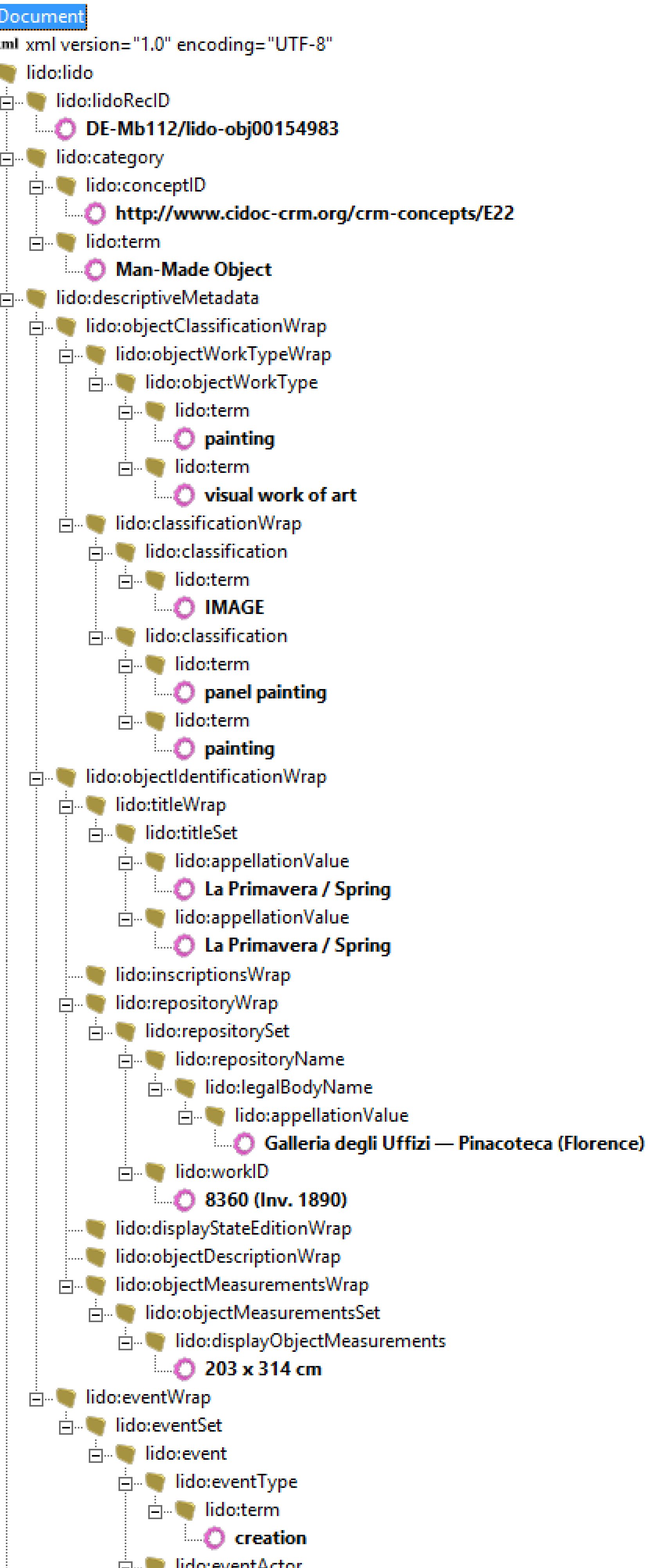
4.3.2 Test cases

In this section we are providing a sample of the tests which have been conducted. We observe the results coming from the schema matching process which affect the result of the mapping process. We have used the Learning method that we have implemented in order to improve the Recall of the system.

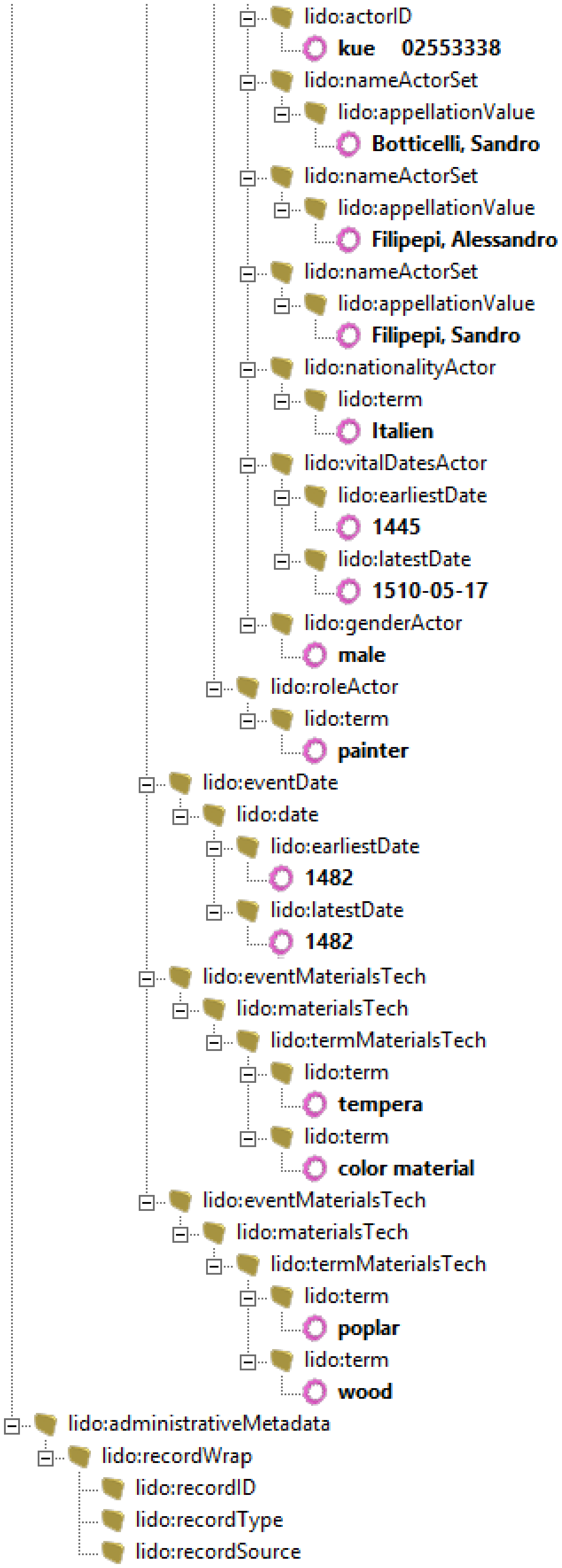


```
Document
  xml version="1.0" encoding="UTF-8"
  cdwalite
    descriptiveMetadata
      objectWorkTypeWrap
        objectWorkType
          oil painting (visual work)
      titleWrap
        titleSet
          title
            Autumn: On the Hudson River
      displayCreator
      indexingCreatorWrap
        indexingCreatorSet
          nameCreatorSet
            nameCreator
              Jasper Francis Cropsey
          nationalityCreator
                      American
          vitalDatesCreator
                      1823-1900
          roleCreator
            painter
      displayMaterialsTech
      displayCreationDate
            1860
      indexingDatesWrap
        indexingDatesSet
          earliestDate
            1860
          latestDate
            1860
      locationWrap
        locationSet
          locationName
            National Gallery of Art (Washington, DC
      descriptiveNoteWrap
        descriptiveNoteSet
          descriptiveNote
                      This monumental view of th
    administrativeMetadata
```

Figure 4.4 CDWA record



```
Document
  xml version="1.0" encoding="UTF-8"
  lido:lido
    lido:lidoRecID
      DE-Mb112/lido-obj00154983
    lido:category
      lido:conceptID
        http://www.cidoc-crm.org/crm-concepts/E22
      lido:term
        Man-Made Object
    lido:descriptiveMetadata
      lido:objectClassificationWrap
        lido:objectWorkTypeWrap
          lido:objectWorkType
            lido:term
              painting
            lido:term
              visual work of art
        lido:classificationWrap
          lido:classification
            lido:term
              IMAGE
          lido:classification
            lido:term
              panel painting
            lido:term
              painting
      lido:objectIdentificationWrap
        lido:titleWrap
          lido:titleSet
            lido:appellationValue
              La Primavera / Spring
            lido:appellationValue
              La Primavera / Spring
        lido:inscriptionsWrap
        lido:repositoryWrap
          lido:repositorySet
            lido:repositoryName
              lido:legalBodyName
                lido:appellationValue
                  Galleria degli Uffizi — Pinacoteca (Florence)
            lido:workID
              8360 (Inv. 1890)
        lido:displayStateEditionWrap
        lido:objectDescriptionWrap
        lido:objectMeasurementsWrap
          lido:objectMeasurementsSet
            lido:displayObjectMeasurements
              203 x 314 cm
      lido:eventWrap
        lido:eventSet
          lido:event
            lido:eventType
              lido:term
                creation
            lido:eventActor
```



```
              lido:actorID
                kue    02553338
              lido:nameActorSet
                lido:appellationValue
                  Botticelli, Sandro
              lido:nameActorSet
                lido:appellationValue
                  Filipepi, Alessandro
              lido:nameActorSet
                lido:appellationValue
                  Filipepi, Sandro
              lido:nationalityActor
                lido:term
                  Italien
              lido:vitalDatesActor
                lido:earliestDate
                  1445
                lido:latestDate
                  1510-05-17
              lido:genderActor
                male
            lido:roleActor
              lido:term
                painter
        lido:eventDate
          lido:date
            lido:earliestDate
              1482
            lido:latestDate
              1482
        lido:eventMaterialsTech
          lido:materialsTech
            lido:termMaterialsTech
              lido:term
                tempera
              lido:term
                color material
        lido:eventMaterialsTech
          lido:materialsTech
            lido:termMaterialsTech
              lido:term
                poplar
              lido:term
                wood
lido:administrativeMetadata
  lido:recordWrap
    lido:recordID
    lido:recordType
    lido:recordSource
```

Figure 4.5 LIDO source record.

In this test we tried to map a CDWA record to the CIDOC –CRM. In the mapping memory we re-used the mapping file of a LIDO record that was mapped in the past. So, we performed a schema matching process between the two XML document files.

The first XML file is a sample CDWA example and describes an oil painting.

The second example is a sample LIDO record that describes a painting "La Primavera".

We used the metrics referred above, Precision, Recall and F-measure. We noticed that Recall reached 1.0 and precision reached 0.66.

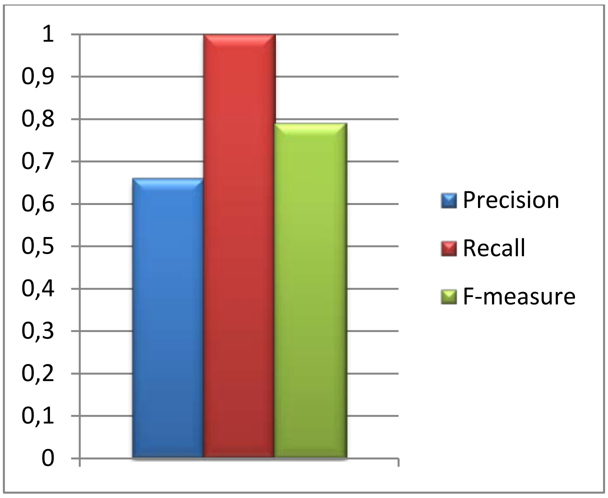


Figure 4.6 Results (CDWA-LIDO)

Based on the crosswalks found by the schema matching process the following suggested paths were displayed to the user.

In table 4.3 we output some of the suggested paths found by the mapping tool.

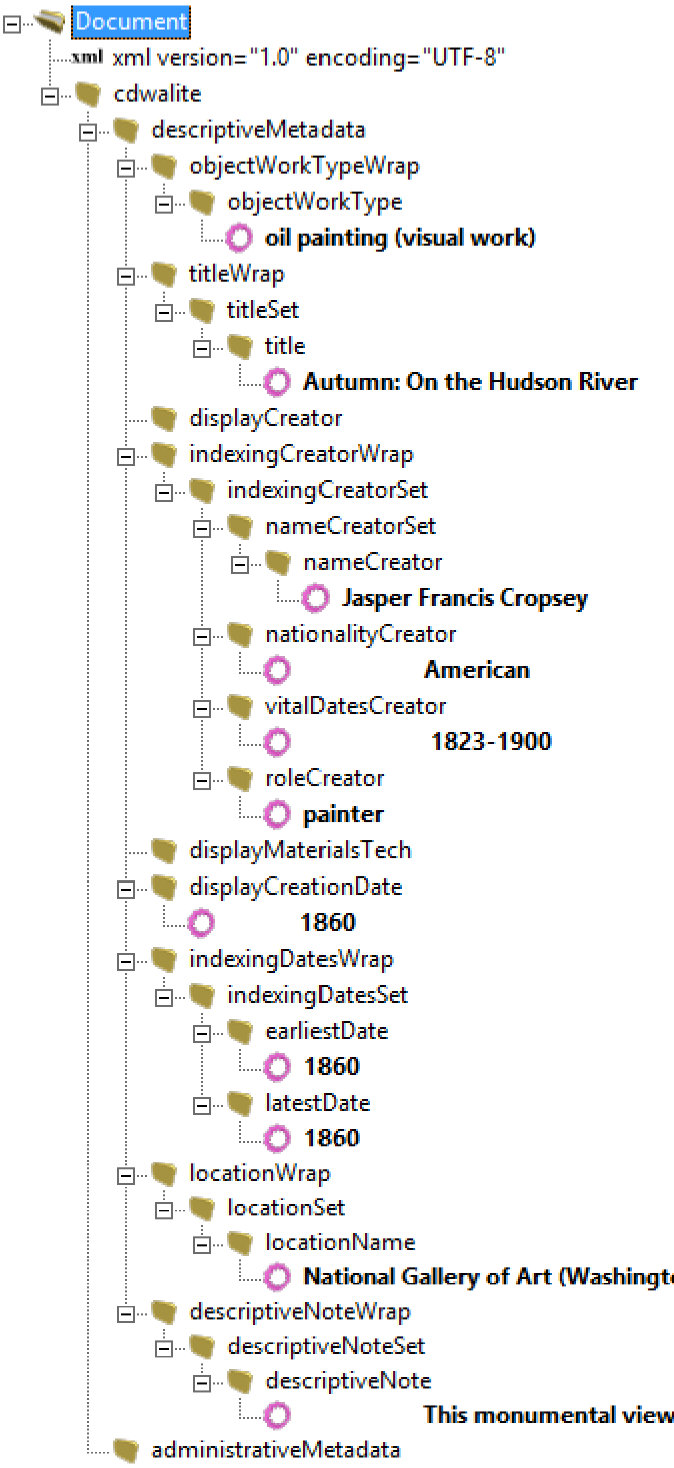
	XPATH	**CIDOC target paths suggested by the tool:**
SOURCE DOMAIN	**/cdwalite**	E22_Man-Made_Object → P2_has_type →E55_Type ✓
SOURCE RANGE	**/cdwalite/descriptiveMetadata/objectWorkTypeWrap/objectWorkType**	
SOURCE DOMAIN	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/nameCreatorSet**	E39_Actor→ P11i_participated_in→ E67_Birth, P4_has_time-span→ E52_Time-Span ✓
SOURCE RANGE	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/vitalDatesCreator** ***(Birth date of actor)***	
SOURCE DOMAIN	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/nameCreatorSet**	E39_Actor→P107i_is_current_or_former_member_of→ E74_Group ✓
SOURCE RANGE	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/nationalityCreator**	

SOURCE DOMAIN	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/nameCreatorSet**	E5_Event→ P11_had_participant→ E39_Actor→ P131_is_identified_by→ E82_Actor_Appellation ✓ ***modified by the user)*** → E39_Actor→ P131_is_identified_by→ E82_Actor_Appellation
SOURCE RANGE	**/cdwalite/descriptiveMetadata/indexingCreatorWrap/indexingCreatorSet/nameCreatorSet/nameCreator**	
SOURCE DOMAIN	/cdwalite/descriptiveMetadata/indexingDatesWrap	**1)** E5_Event→ P4_has_time-span→ E52_Time-Span→ P79_beginning_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal ***(accepted and modified by the user)*** ✓ E65_Creation→ P4_has_time-span→ E52_Time-Span→ P79_beginning_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal **2)** E5_Event→P4_has_time-span→ E52_Time-Span → P80_end_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal ✗
SOURCE RANGE	/cdwalite/descriptiveMetadata/indexingDatesWrap **/earliestDate**	
SOURCE DOMAIN	/cdwalite/descriptiveMetadata/indexingDatesWrap	**1.** E5_Event→ P4_has_time-span→ E52_Time-Span→ P79_beginning_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal ✗
SOURCE RANGE	/cdwalite/descriptiveMetadata/indexingDatesWrap /latestDate	

		2. E5_Event→P4_has_time-span→ E52_Time-Span → P80_end_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal ***(accepted and modified by the user)*** ✓ E65_Creation→ P4_has_time-span→ E52_Time-Span → P80_end_is_qualified_by→ http://www.w3.org/2000/01/rdf-schema#Literal
SOURCE DOMAIN	**/cdwalite**	E22_Man-Made_Object → P1_is_identified_by →E41_Appellation ✓
SOURCE RANGE	**/cdwalite/descriptiveMetadata/titleWrap/titleSet/title**	

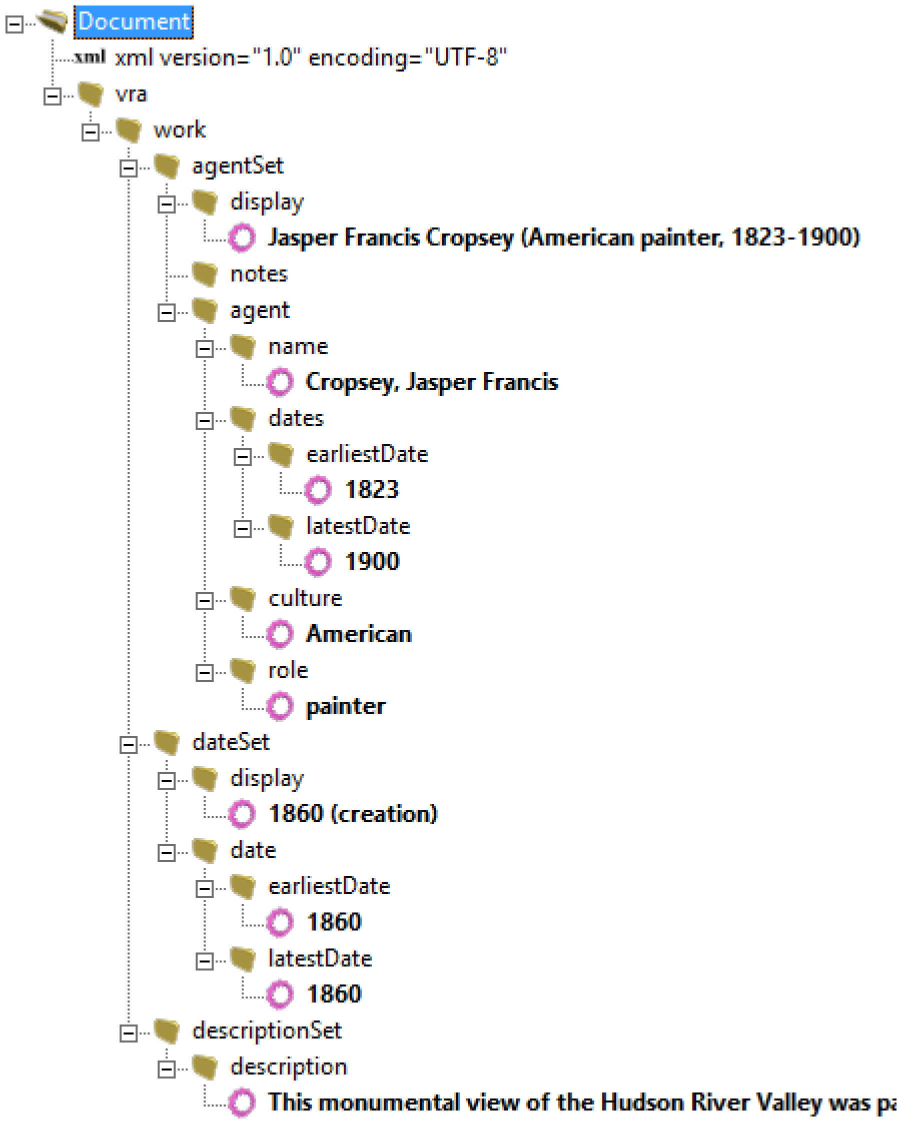
Table 4.3 Examples of suggested target paths

The results proved to be very good cause CDWA and LIDO are relative metadata standards and belong to the same domain.



```
Document
  xml xml version="1.0" encoding="UTF-8"
  cdwalite
    descriptiveMetadata
      objectWorkTypeWrap
        objectWorkType
          oil painting (visual work)
      titleWrap
        titleSet
          title
            Autumn: On the Hudson River
      displayCreator
      indexingCreatorWrap
        indexingCreatorSet
          nameCreatorSet
            nameCreator
              Jasper Francis Cropsey
          nationalityCreator
                      American
          vitalDatesCreator
                      1823-1900
          roleCreator
            painter
      displayMaterialsTech
      displayCreationDate
              1860
      indexingDatesWrap
        indexingDatesSet
          earliestDate
            1860
          latestDate
            1860
      locationWrap
        locationSet
          locationName
            National Gallery of Art (Washington, DC, United States)
      descriptiveNoteWrap
        descriptiveNoteSet
          descriptiveNote
                      This monumental view of the Hudson River V
    administrativeMetadata
```

Figure 4.7 CDWA source record



```
Document
  xml xml version="1.0" encoding="UTF-8"
  vra
    work
      agentSet
        display
          Jasper Francis Cropsey (American painter, 1823-1900)
        notes
        agent
          name
            Cropsey, Jasper Francis
          dates
            earliestDate
              1823
            latestDate
              1900
          culture
            American
          role
            painter
      dateSet
        display
          1860 (creation)
        date
          earliestDate
            1860
          latestDate
            1860
      descriptionSet
        description
          This monumental view of the Hudson River Valley was pa
```

Figure 4.8 VRA source record

In this test we tried to map a CDWA record to the CIDOC –CRM. In the mapping memory we re-used the mapping file of a VRA record that was mapped in the past. So, we performed a schema matching between the two XML document files. The results are depicted below. The precision reached 0.6 and the Recall reached 1.

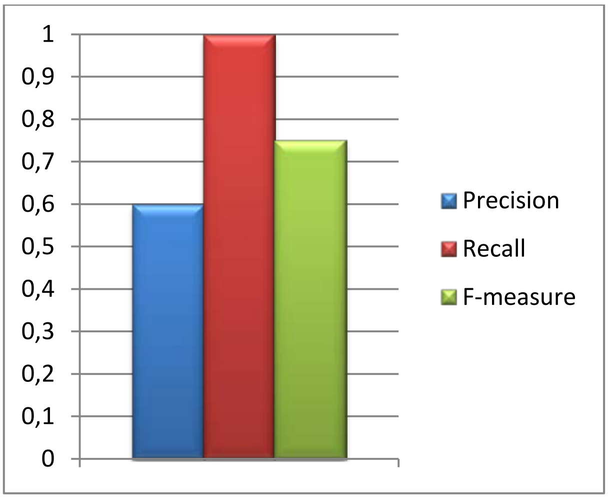
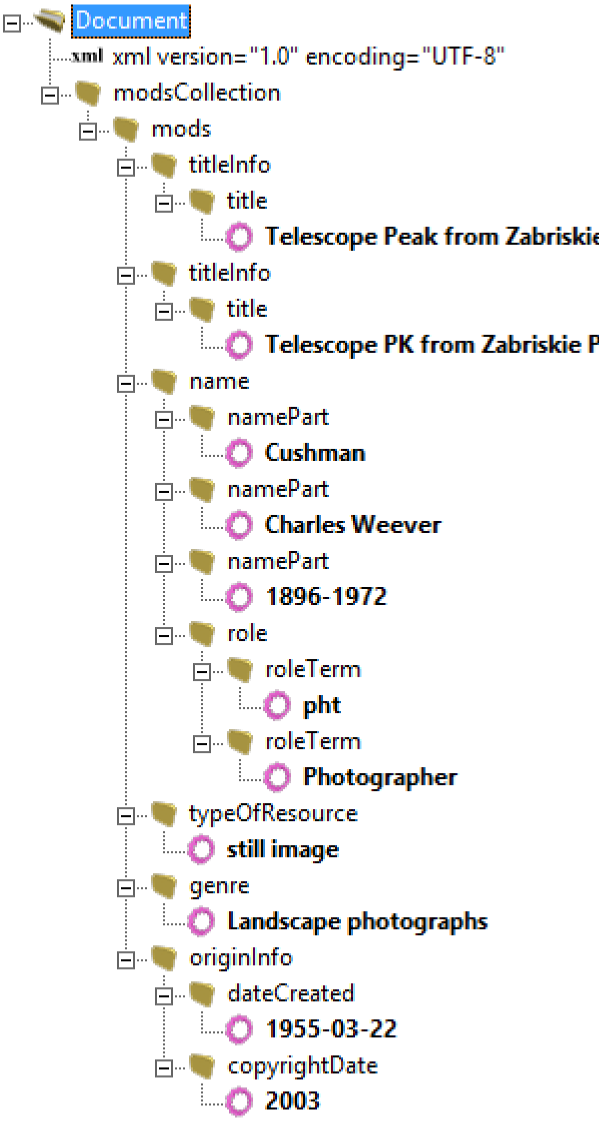
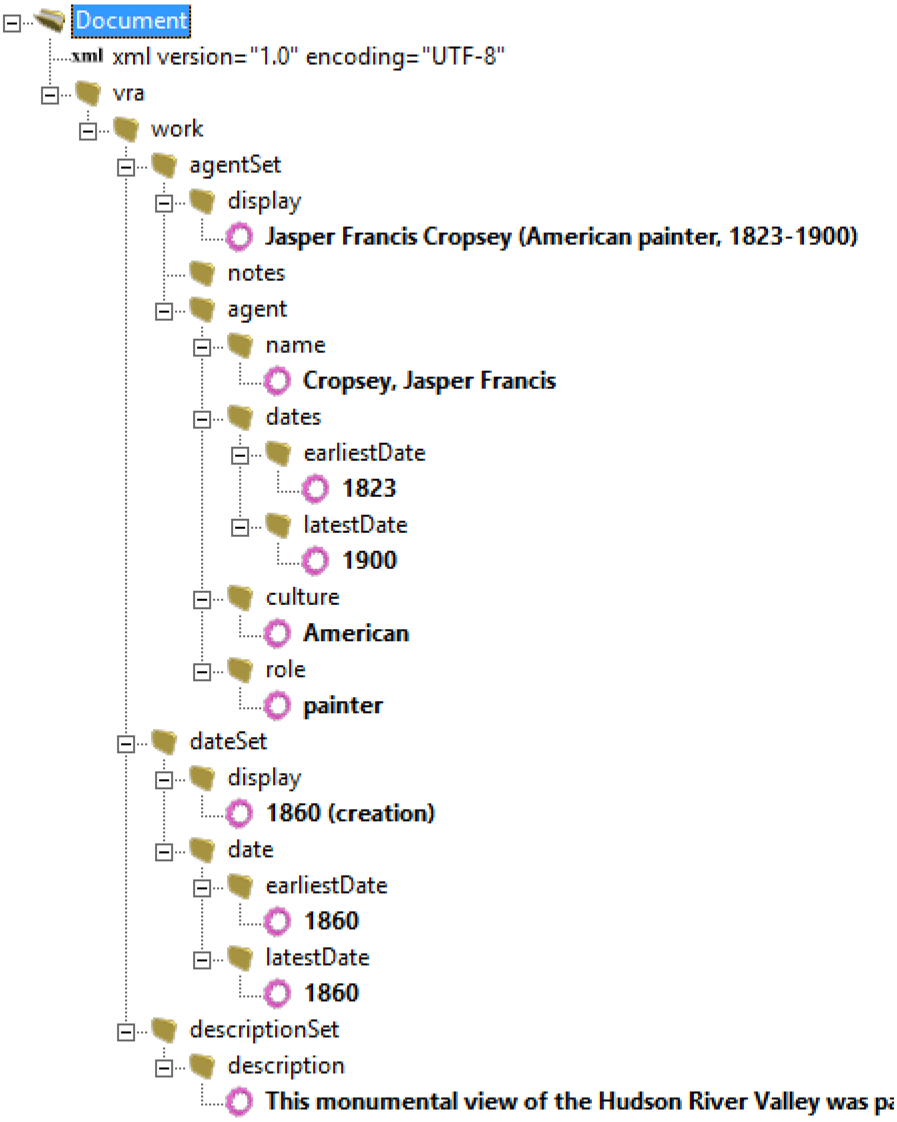


Figure **4.9** Results (CDWA-VRA)



```
Document
  xml xml version="1.0" encoding="UTF-8"
  modsCollection
    mods
      titleInfo
        title
          Telescope Peak from Zabriskie
      titleInfo
        title
          Telescope PK from Zabriskie P
      name
        namePart
          Cushman
        namePart
          Charles Weever
        namePart
          1896-1972
        role
          roleTerm
            pht
          roleTerm
            Photographer
      typeOfResource
        still image
      genre
        Landscape photographs
      originInfo
        dateCreated
          1955-03-22
        copyrightDate
          2003
```

Figure 4.10 MODS source record



```
Document
  xml xml version="1.0" encoding="UTF-8"
  vra
    work
      agentSet
        display
          Jasper Francis Cropsey (American painter, 1823-1900)
        notes
        agent
          name
            Cropsey, Jasper Francis
          dates
            earliestDate
              1823
            latestDate
              1900
          culture
            American
          role
            painter
      dateSet
        display
          1860 (creation)
        date
          earliestDate
            1860
          latestDate
            1860
      descriptionSet
        description
          This monumental view of the Hudson River Valley was pa
```

Figure 4.11 VRA source record

In this test we tried to map a MODS record to the CIDOC –CRM. In the mapping memory we re-used the mapping file of a VRA record that was mapped in the past. So, we performed a schema matching between the two XML document files.

The first XML file is a sample example of the MODS user guide example [96] and describes a digitized photograph.

The second example [97] is a VRA record that describes a painting.

The results from the schema matching tool were as followed.

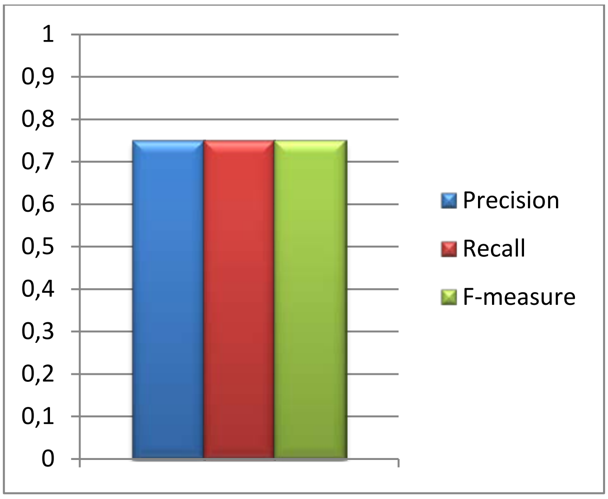


Figure 4.12 Results (MODS-VRA)

In this section, we analyze a sample of the tests that we have conducted. In our tests we observed that the quality of the schema matching results affected the precision and the recall of the semi-automatic mapping tool.

[96] MODS example: http://www.loc.gov/standards/mods/v3/mods-userguide-examples.html#digitized_photograph

[97] VRA Example: http://www.vraweb.org/projects/vracore4/example026.html

Chapter 5

5 Conclusions and Future Work

Conclusions and Future Work

The implementation of a semi-automatic mapping tool that maps a source schema to a target schema ontology supporting a Graphical User Interface and which implies a re-use strategy in order to suggest mappings to the user based on the mapping files stored in our mapping memory (Mapping Suggestion Module) with the assistance of a schema matching tool (Schema Matcher), is described.

The mapping memory is a repository containing mapping files of already mapped XML document files. A mapping file is written in a specific mapping description language and describes the mappings between the elements in the source schema and elements in the target schema. A mapped XML file is a document file the elements of which have been already mapped to the elements in the target schema ontology. The system performs a schema matching between a not mapped XML file and a mapped XML file. Correspondences/Crosswalks found by the schema matching are used by the mapping suggester (an essential component in the system) in order to suggest mappings to the user. User feedback is taken into consideration during the mapping process.

The results of the tests were very satisfactory and supported the usefulness of our system. Of course the results were affected by the output of the schema matching tool and that’s why we tried to tune the schema matching process in order to find the right parameters and choose the best matching strategies that affect the Precision and Recall of the schema matching tool (e.g. matching algorithms, thresholds, user training data, learning methods) . The Learning method that we have implemented improved the Recall of the system. Also, the results of the mapping process were affected by the quality of the mapping files stored in the mapping memory that play a major role in the target path suggestion process and the quantity of the mapping files.

The system and all of its components are written in Java and a Java class library containing their APIs is available for future use by other systems. The components of the proposed system can be integrated in other tools. The system can be used by organizations that have data stored and they would like to map them to target ontologies.

For Future Work, a URI Rule Builder component should be added to the semi-automatic mapping tool in order to form the URI (Uniform Resource Identifiers) rules for each independent node. The URI generation policies for every instance of a target schema class, must be defined, such as for persons, objects, events, place, and formats of time.

Bibliography

1. Alemu, G., Stevens, B., Ross, P., “Towards a conceptual framework for user-driven semantic metadata interoperability in digital libraries: A social constructivist approach.”, New Library World, vol. 113, Iss. 1/2, pp. 38-54. Emerald, 2012.

2. “AMA – Archive Mapper for Archaeology”, EPOCH European Network of Excellence in Open Cultural Heritage, Available from: http://www.epoch-net.org/index.php?option=com_content&task=view&id=222&Itemid=338, Accessed 10 December 2013.

3. Angelopoulou, A., Tsinaraki, C., Christodoulakis, S., “Mapping MPEG-7 to CIDOC/CRM”, Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science, vol. 6966, pp. 40-51. Springer, 2011.

4. “Annocultor”, EPOCH European Network of Excellence in Open Cultural Heritage. Available from: http://semium.org/overview.html, Accessed 10 December 2013.

5. Bellahsene, Z., Duchateau, F., “Tuning for Schema Matching”, Schema Matching and Mapping , pp. 293-316. Springer, 2011.

6. Boeuf, P.L., “Mapping from UNIMARC Bibliographic to CIDOC CRM”, Version 1.0, 2006. Available from: http://www.cidoc-crm.org/docs/UNIMARCB_CRM.zip

7. Bountouri, L., Gergatsoulis, M., “Mapping Encoded Archival Description to CIDOC CRM.” In: Proceedings of the 1st Workshop on Digital Information Management, Corfu, Greece, pp. 8-25. Ionian University, 2011.

8. Bountouri, L., Gergatsoulis, M., Papatheodorou, C., “Mapping EAD to CIDOC CRM”, In: Proceedings of the 21st SIG meeting and 15th FRBR - CIDOC CRM Harmonization meeting Workshop on Conceptual Modelling for Archives, Libraries and Museums, Finland, 2010.

9. “CIDOC Conceptual Reference Model”. Available from: <http://www.cidoc-crm.org/>. Accessed 20 December 2013.

10. Daskalaki, E., “Development and experimental evaluation of an ontology to ontology schema & instance matching system, Master of Science Thesis Computer Science Department, University of Crete”, E-locus University of Crete Institutional Repository, 2011, Available from: http://elocus.lib.uoc.gr/dlib/d/d/b/metadata-dlib-1322814460-437568-20114.tkl

11. David, G., “Understanding Structural and Semantic Heterogeneity in the Context of Database Schema Integration”, In: Proceedings of the 6th Conference in the Dept. of Computing (Journal of the Dept. of Computing), Iss. 4, pp. 29-44. 2005.

12. “Delving SIP-Creator”, Delving open-source solutions, Available from: http://www.delving.eu/the-delving-platform/sip-creator.Accessed: 1st October 2013.

13. “Delving Culture-hub”, Delving open-source solutions, Available from: http://www.delving.eu/the-delving-platform/culture-hub. Accessed 1st October 2013.

14. Do, H.H., Melnik, S., Rahm, E., “Comparison of Schema Matching Evaluations”, Web, Web-Services, and Database Systems, vol. 2593, pp. 221 – 237. Springer, 2003.

15. Doerr M., “Mapping of the Dublin Core Metadata Element Set to the CIDOC CRM”, Technical Report 274, ICS-FORTH, 2000. Available from: http://www.cidoc-crm.org/docs/dc_to_crm_mapping.pdf

16. Doerr, M., “Mapping a Data Structure to the CIDOC Conceptual Reference Model”, ICS-FORTH, Heraklion, Crete, 2002. Available from: http://www.cidoc-crm.org/docs/mapping.ppt

17. Doerr, M., “Mapping of the AMICO data dictionary to the CIDOC CRM”, Technical Report FORTH-ICS/TR-288, 2000. Available from: http://www.cidoc-crm.org/docs/mappingamicotocrm.rtf

18. Doerr, M.,”The CIDOC CRM, an Ontological Approach to Semantic Interoperability of Metadata”, AI Magazine, vol. 24, pp. 75-92, 2003.

19. Doerr, M., Dionissiadou, I.,“Data Example of the CIDOC Reference Model - Epitaphios GE34604 -“, 2007. Available from: http://www.cidoc-crm.org/docs/epitafios1.htm

20. Doerr, M., Felicetti, A., “A Reference model for Data Mapping Tools”, The CIDOC Conceptual Reference Model, FORTH. 2012. Available from: http://www.cidoc-crm.org

21. Doerr, M., Iorizzo, D., “The Dream of a Global Knowledge Network – A New Approach” , Journal on Computing and Cultural Heritage, vol.1, Iss 1, pp. 1-23. ACM, 2008.

22. Duchateau, F., Bellahsene, Z., Coletta, R., "A Flexible Approach for Planning Schema Matching Algorithms", On the Move to Meaningful Internet Systems: OTM 2008, vol. 5331, pp. 249-264. Springer, 2008.

23. Eide, Ø., Felicetti, A., Ore, C.E., Andrea, A.D., Holmen, J., "Encoding Cultural Heritage Information for the Semantic Web". In: Proceedings of the EPOCH Conference on Open Digital Cultural Heritage Systems, pp. 1-7,2008.

24. Eide, Ø., Ore, C-E.," Mapping of TEI to CIDOC-CRM", 2007. Available from: http://www.edd.uio.no/artiklar/tekstkoding/tei_crm_mapping.html

25. "European Network of Excellence in Open Cultural Heritage (EPOCH)", Available from:
http://www.epoch-net.org/index.php?option=com_content&task=view&id=222&Itemid=338.
Accessed 10 December 2013.

26. Felicetti, A., Hubert, M., "Semantic Maps and Digital Islands: Semantic Web technologies for the future of Cultural Heritage Digital Libraries", In: Proceedings of the 5th European Semantic Web Conference , pp. 51-62, Tenerife, Spain, 2008.

27. Gaitanou, P., Bountouri, L., Gergatsoulis, M., "Automatic Generation of Crosswalks through CIDOC CRM.", Metadata and Semantics Research, vol. 343, pp. 264-275. Springer, 2012.

28. Gaitanou, P., Gergatsoulis, M., "A Semantic Mapping of VRA Core 4.0 to the CIDOC Conceptual Reference Model", In: Proceedings of Metadata and Semantic Research 5th International Conference, MTSR 2011, Izmir, Turkey, Communications in Computer and Information Science, vol. 240, pp. 387-399. Springer, 2011.

29. Gaitanou, P., Gergatsoulis, M., “Mapping VRA Core 4.0 to the CIDOC/CRM ontology”, In: Proceedings of the 1st Workshop on Digital Information Management, Corfu, Greece, pp. 26-38. Ionian University, 2011.

30. Gal, A., “Enhancing the Capabilities of Attribute Correspondences”, Schema Matching and Mapping, pp. 53-73. Springer, 2011.

31. Généreux,M., Niccolucci, F.,“Extraction and mapping of CIDOC-CRM encodings from texts and other digital formats”, In: Proceedings of the 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST, 2006.

32. Geser, G., “STERNA (Semantic Web- Based Thematic European Reference Network Application)”, Technology Watch Report, January 2009, Available from: http://www.salzburgresearch.at/wp-content/uploads/2010/10/sterna_tech_watch_report_layoutiert_web.pdf

33. Kakali, C., Lourdi, I., Stasinopoulou, T., Bountouri, L., Papatheodorou, C., Doerr, M., Gergatsoulis, M., “Integrating Dublin Core metadata for cultural heritage collections using ontologies”, In: Proceedings of the International Conference on Dublin Core and Metadata Applications, pp. 128-139, Singapore, 2007.

34. Kalaitzaki, M., : “Design and implementation of a system for semantic schema matching. Master of Science Thesis Computer Science Department, University of Crete”, 2009. Available from:
http://elocus.lib.uoc.gr/dlib/6/5/5/metadata-dlib-4d555a739dd11f5df7c7801d0be78ef8_1275560441.tkl

35. Knoblock, C., “Schema matching”, Available from: http://www.isi.edu/integration/courses/csci548_2008/slides08/Matching.pdf, 2008, Accessed 10 March 2013.

36. Knoblock, C.A., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P., "Semi-Automatically Mapping Structured Sources into the Semantic Web", The Semantic Web: Research and Applications, vol 7295, pp. 375-390.Springer, 2012.

37. Kondylakis, H., Doerr, M., Plexousakis, D., "Mapping Language for Information Integration", Technical Report 385, ICS-FORTH, 2006.

38. Koutraki , M., Doerr , M., "Mapping LIDO v0.7 to CIDOC-CRM v5.0.1", Working paper, FORTH-ICS, 2010. Available from: http://www.cidoc-crm.org/docs/mappings/Mapping_lido_v2.doc

39. Koutrika, G., "Heterogeneity in Digital Libraries: two sides of the same coin", Delos Network of Excellence on digital libraries, Iss 3. DELOS Network of Excellence Newsletter, 2005. Available from: http://www.delos.info/files/pdf/newsletter/delos-newsletter-issue3.pdf

40. Leroi, M. V., Holland, J., "Access to cultural Heritage networks across Europe", ATHENA WP4 Technical Meeting 2010. available online at http://151.12.58.141/athena_mw14/index.php?en/111/events/110/paris-athena-wp4-technical-meeting

41. Li, Y., Liu, D., Zhang,W.,"A Generic Algorithm for Heterogeneous Schema Matching", International Journal of Information Technology, pp. 36-43, 2005.

42. Lourdi, I., Papatheodorou, C., Doerr, M., "Semantic Integration of Collection Description Combining CIDOC/CRM and Dublin Core Collections Application Profile", D-Lib Magazine, vol. 15, number 7/8, 2009.

43. Manakanatas, D., “Design and Implementation of a tool for semi-automated semantic schema matching", Master of Science Thesis Computer Science Department, University of Crete”, E-locus University of Crete Institutional Repository, February 2006.
Available from: http://elocus.lib.uoc.gr/dlib/c/5/7/metadata-dlib-2006manakanatas.tkl

44. “MDA Spectrum CIDOC CRM mapping”, 2003, Available from: http://www.cidoc-crm.org/docs/MDA%20Spectrum_CIDOC_CRM_mapping.pdf

45. Oldman, D., Mahmud, J., Alexiev, V., “ The Conceptual Reference Model Revealed, Quality contextual data for research and engagement: A British Museum case study”, 2013 . Available from https://confluence.ontotext.com/download/attachments/33325240/mapping+manual+for+endpoint+site+draft+0.98a.pdf?version=1&modificationDate=1386147054000/

46. Omelayenko, B., “Porting Cultural Repositories to the Semantic Web”, In: Proceedings of the First Workshop on Semantic Interoperability in the European Digital Library , pp. 14-25, Tenerife, Spain, 2008.

47. Omelayenko, B., ”Semantic Tagging at large scale”, available from http://borys.name/blog/semantic_tagging_of_europeana_data.html, Accessed 19 November 2013.

48. Sheth, A. P., “ Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics, in Interoperating Geographic Information Systems “, Interoperating Geographic Information Systems, vol. 495, pp. 5-29. Springer, 1999.

49. Shvaiko, P., Euzenat, J., “Learning methods” .Chapter In: *Ontology Matching*. pp. 133. Springer, 2007. http://wtlab.um.ac.ir/images/e-library/ontology/Ontology%20Matching.pdf

50. Smith, K., Mork, P., Seligman, L., Rosenthal, A., Morse, M., Wolf, C., Allen, D., and Li, M., ”The role of schema matching in large enterprises”, In: Proceedings of the 4th Biennial Conference on Innovative Data Systems Research (CIDR), 2009.
Available from: http://arxiv.org/ftp/arxiv/papers/0909/0909.1771.pdf

51. Szekely, P., Knoblock, C.A., Yang, F., Zhu, X., Fink, E.E., Allen, R., Goodlander,G., “Connecting the Smithsonian American Art Museum to the Linked Data Cloud”, The Semantic Web: Semantics and Big Data, vol. 7882, pp. 593-607. Springer, 2013.

52. Stasinopoulou, T., Bountouri, L., Kakali, C., Lourdi, I., Papatheodorou, C., Doerr, M., Gergatsoulis, M., “Ontology-based Metadata Integration in the Cultural Heritage Domain”, Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, Lecture Notes in Computer Science,vol. 4822, pp. 165-175. Springer, 2007.

53. Theodoridou, M.,Doerr, M., “Mapping of the encoded Archival Description DTD Element Set to the CIDOC-CRM”, Technical Report 289, ICS-FORTH, 2001, Available from: http://ics.forth.gr/isl/publications/paperlink/ead.pdf

54. Walkowska, J., Werla, M., “Advanced Automatic Mapping from Flat or Hierarchical Metadata Schemas to a Semantic Web Ontology”, Theory and Practice of Digital Libraries, Lecture Notes in Computer Science, vol. 7489, pp 260-272. Springer, 2012.

55. Walkowska, J., Werla, M., “Mapping from Flat or Hierarchical Metadata Schemas to a Semantic Web Ontology”, Available from: http://www.cidoc-crm.org/mapping_technology/jMet2Ont.pdf