

A Framework for Transforming Archaeological Databases to Linked Ontological Datasets

Yi Hong, Monika Solanki,

Department of Computer Science
University of Leicester, UK
{yh37, ms491}@le.ac.uk

Lin Foxhall, Alessandro Quercia

School of Archaeology and Ancient History
University of Leicester, UK
{lf4, aq15}@le.ac.uk

Abstract

Traditionally, archaeological experts are used to recording data in relational databases. Although these data storage mechanisms are robust, they are inaccessible to the most popular and powerful web-based search engines. To increase the uptake and usage of archaeological data, these resources need to be openly available and accessible both to humans and machines. In this paper we propose a framework for the transformation of archaeological databases to RDF datasets. The ontology schema we define for the transformation is an extension of the CIDOC-CRM standard vocabulary. We propose DOTL: a transformation language for migrating the datasets. A novel feature of the transformation is the generation of linked data sources. We exemplify our framework using a case study from the Tracing Networks project.

Keywords: Ontologies, DOTL, Transformation., Linked data, Databases

1. Introduction

Within the last couple of years, digital recording and publishing of archaeological data has rapidly replaced the traditional print medium of publication. Indeed, the Internet has contributed largely to the success of this paradigm shift. The benefits of disseminating archaeological data via the web have been highlighted as early as 2001 [11]. Through the World Wide Web, up-to-date information is being made available on demand, for very large and complex datasets of archaeological information. Along with text, visual information such as maps and photographs are also being digitised and widely disseminated.

In most cases the digital representation of artefacts is accompanied by metadata that describes the contextual information associated with it. Archaeological datasets by their very nature are large, complex and often fragmented bodies of information. They record knowledge about artefacts that represent circumstances underlying diverse historical, social, political and economic contexts. Unsurprisingly in most cases, the information spans across several eras. It is also not uncommon to

find data related to an artefact dispersed across several institutions and made available as disparate and diversely published datasets.

The relationships between the individual elements comprising these islands of information/knowledge span across spatial and temporal boundaries amongst others. In order to fully exploit the potential of the web as a dissemination medium, the datasets and their interrelationships need to be rigorously structured. The large number of linkages (cross references) between individual entities, both within and across the datasets, has to be captured using recommended standards for the Web. This is required to ensure their interoperability, while retaining the flexibility needed to develop applications over the data sources. Representing and publishing information in this fashion allows easy navigation, systematic analysis and efficient information retrieval across the vast number of datasets.

The potential impact of Semantic Web [5] on archaeology has been well recognised [9]. Traditionally, archaeological experts are used to recording data in relational databases. This is usually an SQL repository or the so called "Deep web"[4]. These repositories are only

accessible via web-based end points provided by the holders of the data sources. The majority of the rich and contextually relevant data, which would be of benefit a larger community, lies buried underneath layers of application interfaces. Indeed, relational data silos provide scalable storage and efficient query execution mechanism but they are inaccessible to the most popular and powerful web-based search engines.

To increase the uptake and usage of archaeological data, these resources need to be openly available and accessible both to humans and machines. The Semantic Web provides the infrastructure that is necessary for data to be smartly marked up and made available as ontologies. Archaeological data exposed as ontologies immediately open up the domain to the extensive suite of semantic web aware applications such as data browsers and search engines. Vast amounts of fragmented archaeological data could thus be systematically structured and made available to a wider community.

Yet another Semantic Web initiative which is gaining increasing prominence and is emerging as an important paradigm in connecting the web via networks of data rather than hyperlinks of documents is the notion of "Linked data" [18]. From an archaeological perspective, a very useful benefit of linked data is that as the published archaeological data space on the web grows and new data is linked to existing information sources, via the linked data cloud [20] searches over these datasets deliver complete and updated results.

One such project which is expected to generate large volumes of data stored in RDBs (relational databases) is the Leverhulme funded *Tracing Networks* research programme [1]. The project aims to trace the links between people involved in the production, consumption and distribution of material artifacts across and beyond the Mediterranean region. The project encompasses six closely-linked subprojects dealing with artifacts such as loomweights, and pottery, crafts at Tiryns, coinage, punic ceramics and human representations in art.

In this paper we propose a framework for the transformation of archaeological data sources in the project to RDF [16] data models. Relational data sources are first transformed into data objects. These objects are then mapped into ontology instances using an ECA-based [15] scripting language. The paper is structured as follows: Section 2 discusses the problem with existing mapping approaches. Section 3 presents our proposed transformation framework. Section 4 illustrates our prototype implementation. Section 5 discusses related work and Section 6 presents conclusions and future work.

2. The Problem with Mapping

In this section we highlight several problems that arise while undertaking a transformation from RDBs to ontologies.

Relational database modeling concerns the normalisations of data before it can be stored in a RDB. It entails defining a relational database schema that defines attributes for entities, relationships between the entities and their attributes and those between the entities themselves. One of the limitations of defining a schema in such a way is that the relationships are implicitly defined. Tables in an RDB are related via primary and foreign key constraints. This however does not give a clear indication of the relationships. Since the most interesting and useful aspect of an ontology is its ability to capture and explicitly define relationships, this limitation makes the transformation process non trivial.

Secondly, when converting data from a RDB to ontologies, the scripting languages conventionally used provide simplistic mapping rules. Existing frameworks [7, 8] map table names to RDF/OWL classes and columns to properties in the ontology schema. However they provide limited support in terms of what can be mapped. Generally, the mapping specified is one-to-one, i.e., one column mapped to one concept, as illustrated in *Figure 1*.

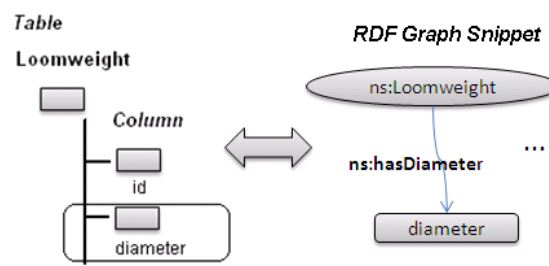


Figure 1: One-to-one mapping example

In most scenarios, the association between columns and properties is far more complex than a simple one-to-one correspondence. This may happen if the domain specific schemas to be used for mapping have been extended from standard vocabularies or those used elsewhere. For example, in this paper, rather than simply generating arbitrary RDF/OWL instances from relational data sources, we would like our ontological instances to conform to a domain specific schema, e.g. CIDOC-CRM [2] for the archaeological domain. It could also be the case that several ontology schemas are used and the data needs to be suitably mapped to more than one property.



Figure 3: Weaving Relationships: Loomweights and cross-cultural networks

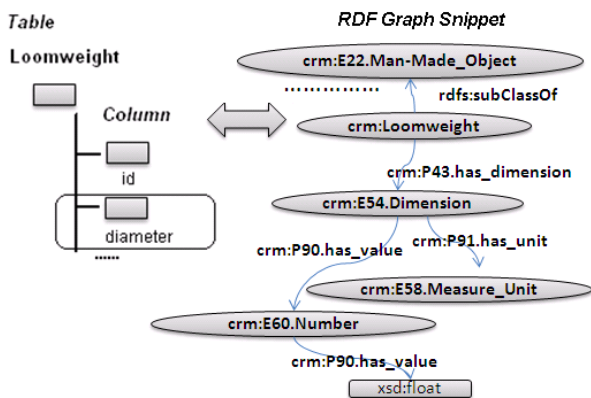


Figure 2: Example of relationship between table and the CIDOC CRM ontology

As an example, we consider data from a subproject of the Tracing Networks research programme. This sub-project investigates loomweights across the Mediterranean and beyond, LBA-3rd c. BCE. A much neglected artefact-type, they provide important information about textile production on the warp-weighted loom, largely a women's activity in many Mediterranean societies. They come in a wide range of sizes, weights and clay fabrics, partly related to the particular cloth

manufactured. Loomweights can reveal social links and personal/group (often feminine) identities. Styles change broadly in harmony over a wide area of the Mediterranean, with many local variations. Types jump both across regions and across cultures (e.g. S. Italy, where indigenous cultures adopt local versions of Greek-style loomweights, but indigenous types also appear in Greek contexts). They may be individualized with stamps, fingerprints, etc., suggesting that they were valued as personal possessions. Stamps and loomweights can be tracked geographically and chronologically (e.g. examples of 4th c. loomweights with 6th c. stamps on them in Metaponto). Some are professionally made; others appear home-made. Systematic study of their manufacture and use over a range of contexts (kilns, houses, graves, sanctuaries) will illuminate textile production across the Mediterranean world, adding new insights on identities and social relationships, especially networks of women.

A loomweight is characterised by attributes such as number of holes, stamps (impressions) thickness and diameter. Figure 3 illustrates various types of loomweights.

The ontology schema used for the transformation has been extended from CIDOC-CRM. This is illustrated in Figure 2. In this figure, it can be seen that the column

diameter of the RDB table cannot be mapped in a straightforward manner, i.e., as a datatype property, in the schema. In order to specify a relationship between *diameter* and the concept *Loomweight* we have to create several intermediate instances of concepts defined by CIDOC-CRM. These instances have to be contextually related to each other in order to ensure that Loomweights are assigned correct diameter values.

3. Transformation Framework

To address the problems of mapping data sources, we propose a new framework for transforming archaeological RDBs to ontological datasets that are extensions of CIDOC-CRM, an ISO standard for the integration of cultural and heritage information. The transformation workflow, as depicted in *Figure 1*, consists of three main steps:

- (1) ORM Reverse Engineering.
- (2) ECA Rule-based Transformation.
- (3) Ontology Instance Generation.

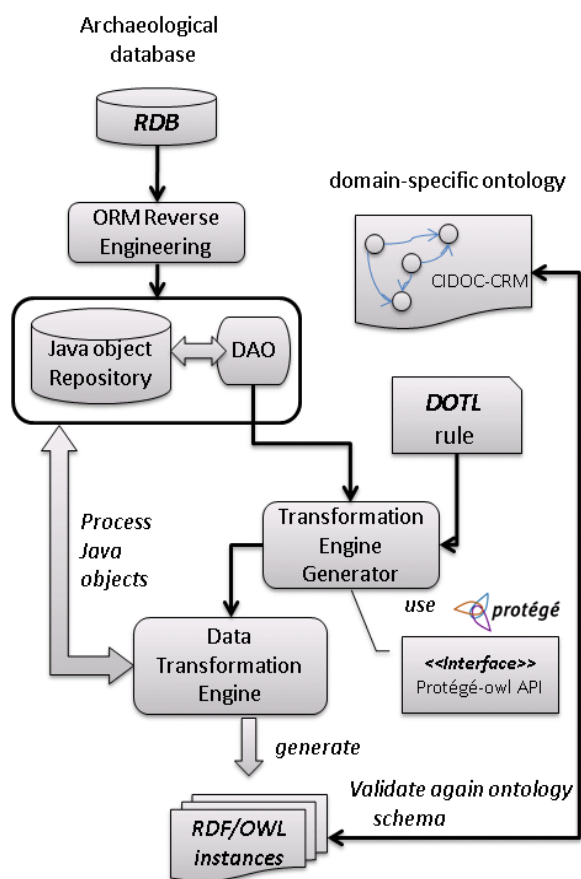


Figure 4: Transformation Framework.

It is worth noting that although in this paper, we use the framework to transform archaeological databases, our approach is generic and can be used for the transformation of data from most popular RDBs. In the fol-

lowing sections, we briefly explain the three steps involved in the transformation.

3.1. ORM Reverse Engineering

In this framework we use the ORM reverse engineering technique [13] to extract concepts and relationships from the RDB. The ORM (Object-Relational Mapping) technique is commonly used for storing persistent object-oriented data within heterogeneous RDBs. On the other hand, the ORM Reverse Engineering technique is used to extract the existing tables, columns, relationships (inc. primary key, foreign key, join etc) and index from a relational database to object-oriented data structures. It has the capability of representing an RDB table as data structures or “classes”. Therefore, records in the table can be instantiated as data objects, which can be easily manipulated and processed using Object-Oriented Programming (OOP).

Since we have a huge amount of data distributed across several different legacy RDBs, e.g., MySQL and MS Access, amongst others, the transformation framework must be designed to be able to flexibly connect to various databases to retrieve the data. One of the most popular and compatible data persistence solutions is the open source Hibernate ORM [12] framework. In our proposed approach, the Hibernate ORM Reverse Engineering tool is used to convert database records into Java objects and DAO (Data Access Object) which act as an interface to access the Java objects. Technical details and database compatibility issues in this regards are further discussed in section 4.

3.2. ECA Transformation Rules

After converting database records into object-oriented data, the next and the most important step is to define transformation rules for these objects. The conventional mapping mechanism defines mapping rules in their own language such as D2RQ[7] and Virtuoso[8]. This approach basically maps tables to RDF/OWL classes and columns to properties. Such an approach works perfectly fine with straight forward one- to-one mapping we discussed in *Figure 1*.

However this approach lacks the capability of expressing complex non-linear relationship as shown in *Figure 2*. The figure illustrates the type of mapping which is very common when transforming databases to sophisticated target ontologies such as CIDOC-CRM.

To facilitate the transformation, we developed an ECA-based [15] (Event-Condition-Action) textual transformation language *DOTL — Database Ontology Transformation Language*. Existing mapping mechanisms merely specify one-to-one mappings at schema level and work only on simple schema models, the objective is to extract data from the database as ontologies without any emphasis on the resulting structure.

On the other hand, the ECA mechanism allows us to define transformations based on complex conditions. In addition, we can define a set of actions that can generate ontological instances using our expressive CIDOC-CRM extended ontology schema. Our approach is very flexible and can be used to generate ontological instances based on large and expressive ontological schemas.

The fundamental construct of a *DOTL* transformation rule is of form:

On Event if Condition Do Action

A basic *DOTL* rule consists of three parts:

- (1) The *event* part specifies the triggers of the transformation rule, usually the occurrence of an object of a specific class;
- (2) The *condition* part is a logical expression, which checks the pre-condition of the action to be carried out, the default conditions being “if undefined”;
- (3) The *action* part usually consists of a series of creation of new ontology instances, properties and other corresponding operations.

```
foreach item:Loomweight do{
  createOWLInstance loomweight
  type "crm:loomweight"
  URI "crm:loomweight_" + item.id

  onFeature item.diameter {

    createRDFInstance dimension
    type "crm:E54.Dimension"
    URI "crm:E54.Dimension_" + item.id

    createRDFInstance number
    type "crm:E54.Number"

    createRDFInstance cm
    type "crm:E58.Measurement_unit"
    URI "crm:E58.Measurement_unit_cm"+item.id

    createPrimitive integer
    type "Integer"
    value item.diameter

    createOWLProperty ("tn:hasMaxHoleDiameter",loomweight,dimension)
    createRDFProperty ("crm:P90.has_value",dimension,number)
    createRDFProperty ("crm:has_FloatNumber",number,integer)
    createRDFProperty ("crm:P91.has_unit",dimension,cm)
  }
}
```

Listing 1: *DOTL transformation rules*

The *DOTL* code snippet in *Listing 1* outlines the mapping rules for the *Loomweight* table when the transformation procedures illustrated in *Figure 3* are applied to the RDB sources: for each *Loomweight* element in the object collection, create a corresponding RDF *Loomweight* individual; if the value of attribute “*diameter*” is not null, then perform a sequence of operations as listed in *Listing 1*, such as creating a new instance of intermediate class *Dimension*, specifying the URI pattern, establishing a link between *Dimension* and *Loomweight* instance, assigning value to appropriate data type property.

3.3. Instance Generation as Linked Data

Once the transformation rules are specified, the final step is to generate ontological instances as linked data. This is a two step process. Rather than a compiler, our framework includes an important component, *Transformation Engine Generator (TEG)* to get the executable code. To initiate the transformation, the generator takes as input the *DOTL* rules and the DAO. As an output, *TEG* produces another component, the *Data Transformation Engine (DTE)*. The engine is then responsible for converting the data accessed through the DAO to ontological instances as per the schema defined for the ontologies.

DTE generates loomweight instances as linked data. We currently link to the DBpedia [21] and Geonames [22] datasets. *Listing 2* lists a loomweight instance with data linked from Dbpedia and Geonames.

```
<rdf:RDF
  xmlns:crm="http://www8.informatik.uni-erlangen.de/INMDS/Services/cidoc-crm/s"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:tn="http://www.tracinetnetworks.ac.uk/ontology/loomweight.owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.tracinetnetworks.ac.uk/ontology/loomweight.owl#"
  <Loomweight rdf:ID="loomweight Nayla Pantanello 255">
  <crm:P3.has_note rdf:resource="#note_255"/>
  <has_loomweight_description rdf:datatype="xsd:string">
  Loom weight
  </has_loomweight_description>
  <crm:P45.consists_of rdf:resource="#Terracotta"/>
  <crm:P2.has_type rdf:resource="#Lentoid"/>
  <has_item_number rdf:resource="#item_number_255"/>
  <has_thickness rdf:resource="#thickness_255"/>
  <has_lf_note rdf:resource="#lf_note_255"/>
  <has_level rdf:resource="#level_255"/>
  <has_loomweight_square rdf:resource="#square_255"/>
  <has_battuta rdf:resource="#battuta_7.0"/>
  <has_diameter rdf:resource="#diameter_255"/>
  <has_max_hole_diameter rdf:resource="#max_hole_diameter_255"/>
  <has_weight rdf:resource="#weight_255"/>
  <has_hole rdf:resource="#loomweight_hole_255_No_1"/>
  <has_hole rdf:resource="#loomweight_hole_255_No_2"/>
  <has_inventory_no rdf:resource="#Inventory Number 81.0578"/>
  <is_discovered_during_excavation rdf:resource="#Excavation activity_255"/>
  <has_provenience rdf:resource="#place_NW_trench_brown_sand_with_gravel"/>
  <has_related_resources_in>
  <crm:E53.Place rdf:ID="Europe">
  <crm:P89I.contains>
  <crm:E53.Place rdf:ID="Greece">
  <owl:sameAs rdf:resource="http://sws.geonames.org/390903"/>
  <owl:sameAs rdf:resource="http://www.dbpedia.org/resource/Greece"/>
  <crm:P89_falls_within rdf:resource="#Europe"/>
  </crm:E53.Place>
  </crm:P89I.contains>
  <owl:sameAs rdf:resource="http://sws.geonames.org/6255148"/>
  <owl:sameAs rdf:resource="http://www.dbpedia.org/resource/Europe"/>
  </crm:E53.Place>
  <crm:E53.Place rdf:ID="Egypt">
  <owl:sameAs rdf:resource="http://sws.geonames.org/3582161"/>
  <owl:sameAs rdf:resource="http://www.dbpedia.org/resource/Egypt"/>
  <crm:P89_falls_within>
  <crm:E53.Place rdf:ID="Africa">
  <crm:P89I.contains rdf:resource="#Egypt"/>
  <owl:sameAs rdf:resource="http://sws.geonames.org/6255145"/>
  <owl:sameAs rdf:resource="http://www.dbpedia.org/resource/Africa"/>
  </crm:E53.Place>
  </crm:P89_falls_within>
  </crm:E53.Place>
  </has_related_resources_in>
  </Loomweight>
</rdf:RDF>
```

Listing 2: *Loomweight instance as linked data*

Instead of creating a virtual in-memory model on the fly, the framework exports all data to the RDF store or as persistent RDF files.

4. Implementation

A prototype implementation of the framework has been developed in Java. The prototype uses the open source Hibernate Reverse Engineering framework for object/relational mapping, which provides connectivity support for Oracle, DB2, SQL Server, MySQL and most mainstream relational databases. In this way, we are able to retrieve data from various existing archaeological databases as well as integrate data sources in distributed RDBs.

A textual DOTL Editor plugin for Eclipse has also been implemented, which supports syntax coloring, code completion, syntax checking/error markers, and navigation. Formalised EBNF grammar of DOTL is defined in Xtext [14] syntax while the metamodel of the language is described using the EMF [14] (Eclipse Modeling Framework). The DOTL Editor also contains an integrated Java code generator implemented in Xpand [14] for building executable transformation program. Generated Java code invokes Protégé OWL API to perform the generation of RDF/OWL instances.

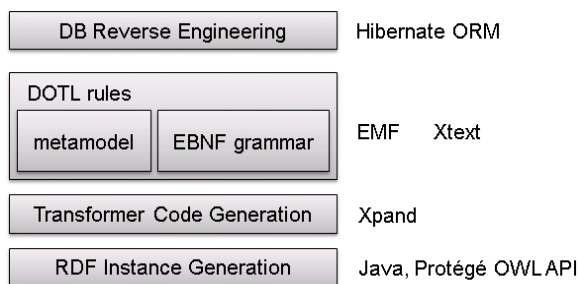
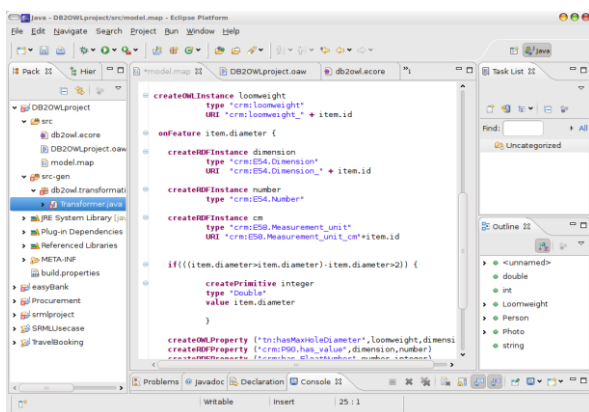


Figure 5: Layers and Implementation Techniques

Figure 5 illustrates the layers of the transformation framework and relevant techniques used during the prototype implementation.

Figure 6: DOTL Editor plugin for eclipse



5. Related work

Since the first step in semantically enabling data is their transformation to ontological instances, varied techniques and approaches have been proposed by different working groups.

A survey of approaches for mapping RDB to RDF is presented in [10]. D2RQ [7] and Virtuoso [8] provide a declarative language to describe mappings between relational database schemata and OWL/RDFS ontologies. The language is limited to specifying basic triple pattern match. An approach closely related to our work is R2O [3]. The authors specify an XML-based language for the transformation. In related archaeology projects [17], the creation of initial mappings between database columns and RDF entities was a manual exercise undertaken with the benefit of domain knowledge from English Heritage. In [6] bespoke tools are provided that guide the data curators through the mapping process, using basic natural language processing. Within the domain of cultural heritage, a closely related domain to archaeology, a large number of projects have implemented diverse transformation mechanism. An example of triplification of cultural heritage data can be found in [23]. In our approach we can specify complex mapping patterns for ontology transformation. RDB schemas can be quite complicated in terms of integrity constraints. Our framework provides support for dealing with many of these constraints, since we build our work on the Hibernate framework.

6. Conclusion and Future work

Little work has been done so far in the Semantic Web community that can motivate archaeologists to adopt their technologies to manage and analysis data. Interesting results have been obtained in the domain of cultural heritage and museums, however most of these efforts have focused on archiving data as ontological datasets rather than performing interesting statistical analysis. Archaeology by its very nature focuses on establishing linkages between past events, places, people and things. The Semantic Web infrastructure therefore serves as a potential solution because of its emphasis on capturing relationships and should be exploited to provide archaeological data management solutions.

In this paper we proposed a transformation framework for migrating large volumes of archaeological data stored in RDBs to ontology based data sets on the Semantic Web. We proposed the ECA-based scripting language DOTL, which allows the specification of complex transformation rules from data objects to ontologies. We discussed a motivating example based on the CIDOC-CRM ontology schema as a case study.

Finally we presented a prototype implementation that illustrates our methodology.

We are currently refining the grammar and semantics to enhance the expressiveness of DOTL to improve the usability of the system. Another ongoing development is to implement a user-friendly graphical modeling environment for the language in GMF (Graphical Modeling Framework) to allow easy creation and editing of transformation rules. The linked data cloud does not currently hold any archaeological datasets, however we are keeping an eye on the cloud for any possible datasets that could be published in the near future. As other subprojects of Tracing Networks start producing their datasets, we will also be linking these with the loom-weight instances.

Acknowledgement. This work was partially supported by the Leverhulme Trust Programme Award “Tracing Networks”.

References

[1] Tracing Networks Research Programme: Craft Traditions in the Ancient Mediterranean and Beyond, <http://www.tracingnetworks.ac.uk>

[2] The CIDOC Conceptual Reference Model
<http://cidoc.ics.forth.gr>

[3] Jesús Barras, Óscar Corcho and Asunción Gómez-pérez. 2004, R2O, An Extensible and Semantically based Database-to-Ontology Mapping Language

[4] Michael K Bergman. The Deep Web: Surfacing Hidden Value.
http://www.brightplanet.com/images/uploads/DeepWebWhitePaper_20091015.pdf

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web, 2001, *Scientific American Magazine*

[6] Leif Isaksen and Kirk Martinez and Nicholas Gibbins and Graeme Earl and Simon Keay. 2009. Linking Archaeological Data, *Computer Applications and Quantitative Methods in Archaeology conference*

[7] Christian Bizer and Andy Seaborne. 2004, D2rq - treating non-rdf databases as virtual RDF graphs. In *ISWC2004 (posters)*.

[8] Orri Erling and Ivan Mikhailov. 2007, RDF support in the Virtuoso DBMS. In *Conference on Social Se-*

mantic Web.

[9] Julian D. Richards. 2006, Archaeology, e-publication and the semantic web. *Antiquity*, (310).

[10] Satya S. Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda and Ahmed Ezzat. 2009, *A Survey of Current Approaches for Mapping of Relational Databases to RDF*.

[11] J. David Scholen. Archaeological data models and web publication using xml. 2001, *Computers and the Humanities*, 35(3).

[12] Red Hat Middleware, 2009, “Relational Persistence for Java and .NET”, www.hibernate.org

[13] Astrova, I., 2004. Reverse Engineering of Relational Databases to Ontologies, *The Semantic Web: Research and Applications*, pp: 327-341, LNCS

[14] Gronback R.C, 2008. Eclipse Modeling Project – A Domain-Specific Language (DSL) Toolkit, pp 277-313, 605-649, Addison-Wesley Pearson Education

[15] James Bailey, Alexandra Poulouvasilis and Peter T. Wood, 2002. An Event-Condition-Action language for XML, pp: 486 - 495, In Proceedings of *The 11th international conference on World Wide Web*

[16] Resource Description Framework (RDF), <http://www.w3.org/RDF>

[17] Binding, Cer, May, Keith and Tudhope, Douglas, 2008, Semantic Interoperability in Archaeological Datasets: Data Mapping and Extraction Via the CIDOC CRM, ECDL '08: *12th European conference on Research and Advanced Technology for Digital Libraries*

[18] Linked Data: Connect Distributed Data across the Web
<http://linkeddata.org>

[19] Christian Bizer and Tom Heath and Tim Berners-Lee, 2009. Linked Data - The Story So Far, *International Journal on Semantic Web and Information Systems*

[20] Linked Data Cloud
http://www4.wiwiw.fu-berlin.de/bizer/pub/lod-datasets_2009-03-05_colored.png

[21] DBpedia, <http://dbpedia.org>

[22] Geoname, <http://sws.geonames.org/>

[23] K. Byrne, 2008, Having Triplets - Holding Cultural Data as RDF, *Information Access to Cultural Heritage: ECDL 2008 Workshop, Aarhus Denmark*