# A Semantic Portal for Fund Finding in the EU: Semantic Upgrade, Integration and Publication of Heterogeneous Legacy Data

Jesús Barrasa Rodríguez, Oscar Corcho[1], and Asunción Gómez-Pérez

Ontology Engineering Group, Departamento de Inteligencia Artificial,
Facultad de Informática, Universidad Politécnica de Madrid, Spain
jbarrasa@eui.upm.es, ocorcho@fi.upm.es, asun@fi.upm.es

**Abstract.** FundFinder is a Semantic Web portal that allows searching for and navigating through information about funding opportunities. This application has been created following a set of techniques and using a set of tools for the upgrade of legacy content to the Semantic Web, including databases and semi-structured documents. This process consists in extracting and populating knowledge from heterogeneous information sources and making it available on the Web.

## 1 Introduction

Nowadays there are several Web portals that contain information related to funding opportunities for different types of organisations and individuals. Examples of such portals in the context of the European Union are CORDIS[2] or the EU's Grants and Loans site[3]. These types of portals are also available at national, regional or local levels in the different EU member states.

One example of such portal, at the regional level, is the public Website of CIDEM[4], which is a non-profit Catalan organisation that aims at improving the region's industrial community networks and at increasing their competitiveness. This Website contains information about funding opportunities gathered, manually and on a daily basis, by CIDEM's staff members from different sources (mainly official publications). Access to this content is provided by standard form-based web pages that allow users to specify some basic search criteria such as the productive sector (Agriculture, Industry, Services, Tourism, Non-profit Organizations, etc.) to which the funding applies, the funding's objective (Technical and Financial Consultancy, Business cooperation, Culture, Energy, Tax incentives, Environment, R&D, Training, etc.), the date of the last update (to get the newest ones), etc. A traditional full text search engine is also provided to ease the search for funding opportunities.

Search interfaces like this one are helpful for basic information retrieval, with questions like *"Give me all funding opportunities in the agriculture sector"* or *"Give*

[1] Currently at the University of Manchester (Oscar.Corcho@manchester.ac.uk).
[2] Community Research and Development Information Service (http://ica.cordis.lu/search/).
[3] http://europa.eu.int/grants/index_en.htm
[4] Centre for Innovation and Business Development (http://www.cidem.com).

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part III, LNAI 4253, pp. 588 – 597, 2006.

*me all funding opportunities containing the words* 'sustainable development'". However, they fall short for dealing with complex queries involving relations between concepts, such as *"Give me all the funding opportunities that can provide a supplement to those aiming at company creation"* or *"Give me all the funding opportunities that are incompatible with funding 651"*. The reason for this is that answering these types of questions requires understanding the meaning of the relations *"provide a supplement"* and *"be incompatible with"*.

Ontologies can provide a shared understanding of such relations and, in general, of most of the terms used in such queries. When ontologies are integrated in Web portals we normally talk, indistinctly, about the terms knowledge portals, semantic portals, community Web portals or Semantic Web portals [7].

In this paper we describe how we have created the Fund Finder application, whose objective is to allow semantic access to the content available in the current CIDEM portal, integrated with content from other heterogeneous sources. In other words, we describe the process of upgrading the current CIDEM portal to the Semantic Web, for which we have used some of the approaches, techniques and tools developed in the context of the project Esperonto. These are:

- A set of domain ontologies (covering the funding domain and other domains related to it.)
- An automatic processor called ODEMapster, capable of transforming information from databases into knowledge bases, according to a declarative mapping description document previously specified.
- An automatic processor called Knowledge Parser, capable of extracting information from semi-structured documents and populating it into a knowledge base, according to a configuration previously specified.
- A publication tool called ODESeW [5], capable of deploying Semantic Web portals with semantic navigation and querying functionalities.

In Section 2 we briefly describe the set of ontologies developed to formalize the Funding domain. Section 3 details how the generation of semantic content from heterogeneous information sources is carried out, providing an overview of the approach followed and of some of the tools involved in the process: the $R_2O$ mapping description language and the ODEMapster processor, on the one hand, and Knowledge Parser, on the other hand. Section 4 focuses on presentation, describing how the semantic content is presented and how it can be queried by final users. Section 5 concludes the paper and describes how to replicate this approach in another domain. A final appendix on the $R_2O$ mapping description language and the ODEMapster processor provides further details about them.

## 2 Ontologies in the Funding Domain

Ontologies are defined as formal, explicit specifications of a shared conceptualization [8]. In the context of our Fund Finder application, ontologies represent the domain of funding opportunities, funding bodies, applicants, organisations, persons, locations, publications, etc. These ontologies can be divided into two layers: the higher one

contains general-purpose highly reusable ontologies (Person, Location, Organization, Official Publication), while the lower one contains specific ontologies specifically related to funding (Funding Opportunity, Funding Body, Applicant). Figure 1 presents these two layers and an inter-ontology relation diagram that summarizes the main relations between these ontologies.

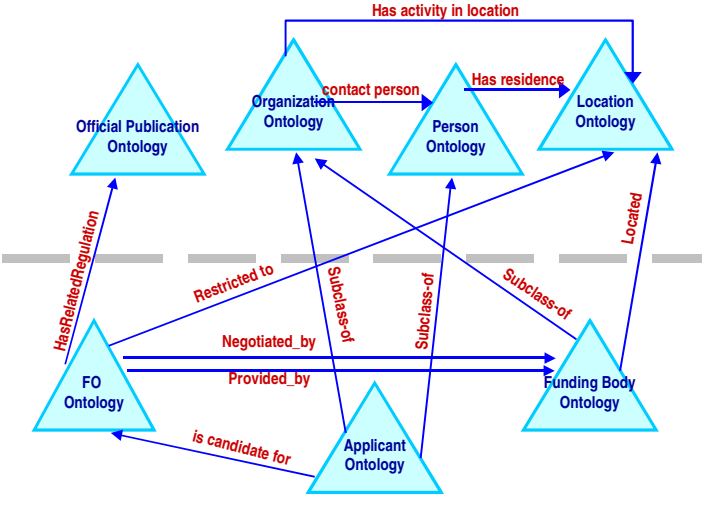


**Fig. 1.** Inter-ontology relationships between the different Fund Finder ontologies

It is not the purpose of this paper to explain in detail the knowledge formalised in these ontologies. We will just say that they have been developed by experts in the domain of funding in the European Union, following the Methontology methodology [6] and using the WebODE ontology engineering workbench [1].

## 3 Automatic Semantic Content Generation

One of the biggest barriers to large-scale deployment of Semantic Web applications is the availability of semantic content [3]. This content can be created by annotating existing information sources, by using different types of annotation techniques and tools, with different degrees of human supervision and annotation accuracy.

In the case of the Fund Finder application, existing content is stored in a relational database, owned by CIDEM and updated on a weekly basis, and in PDF and HTML documents available from several official journal Web sites (Catalan, Spanish, European, etc.). In the following sections we briefly describe how we extract knowledge from the different types of information sources, using the $R_2O$ language and ODE-Mapster for databases and Knowledge Parser for the PDF and HTML documents, and how we populate the funding opportunity ontologies integrating knowledge coming from these different sources.

### 3.1 $R_2O$ and ODEMapster: Database-to-Ontology Mapping

As aforementioned, some of the content to be upgraded to the Semantic Web resides in a legacy database that belongs to CIDEM. This database was developed several years ago with the purpose of being used as a backend for the Web application provided by this organisation. The database is updated manually, on a weekly basis, using different types of information sources as official journals, internal documents, faxes, etc.

Our objective is to be able to access the contents of the database as if they consisted of instances of the domain ontologies defined for our application. However, the process is not straightforward, since although the database schema and the ontologies cover overlapping parts of the domain, the models are usually different (databases are modelled with the objective of being used as data backends for applications while ontologies are modelled with the objective of representing the domain).

The $R_2O$ language [2] has been developed with the purpose of allowing the declarative specification of mappings between a database and a set of ontologies, so that these mappings can be later processed by the ODEMapster processor [2] in order to transform the content of the database into instances of an ontology implemented in a Semantic Web language like RDF Schema or OWL, as shown in Figure 2. The transformation can be done in a batch mode (all the RDF statements the result from applying the mappings are generated and stored somewhere in the application) or on-demand (only the mappings that are relevant for a query are executed when a query is sent to the system).

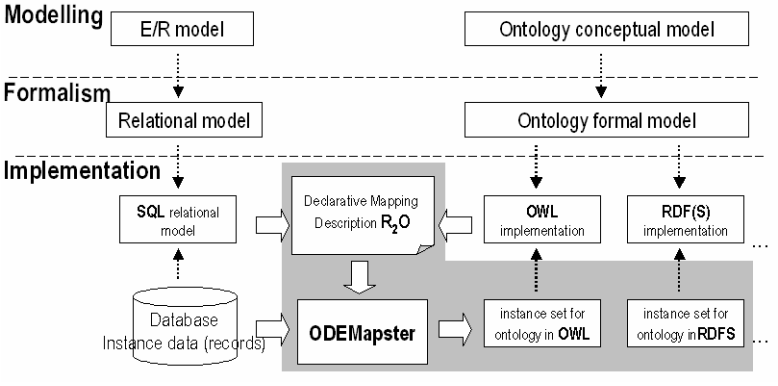


**Fig. 2.** $R_2O$ mapping architecture

$R_2O$ is intended to be expressive enough to describe the semantics of a great range of mappings between relational databases and ontologies. It is independent of the RDBMS, working with any DB implementing the SQL standard.

Because the domains covered by the ontology and the database do not always coincide and because the design modeling criteria used for building the DB are different from those used for ontology creation, the correspondences between their elements will be sometimes straightforward and sometimes difficult. $R_2O$ distinguishes the

following cases in concept transformation: 1) one DB table or view maps one concept in the ontology, 2) one DB table or view is used to instantiate more than one concept in the ontology, but only one instance per concept, 3) one DB table or view is used to instantiate more than one concept in the ontology, but multiple instances of the ontology can be generated.

Furthermore, before generating ontology instances, some standard relational algebraic operations (projection, selection, etc.) usually need to be executed, such as: Direct Mapping, Join/Union, Projection, Selection, or any combination of them.

Finally, the values of the attributes and relations can be filled in directly from the values of the fields in a DB record or after the application of a transformation function, which can affect more than one data field.

Although SQL relational algebra operations cover many cases, there are situations in which some additional transformations might be needed. Examples are more complex operations like natural language processing techniques over text data fields, regular expression matching for dates, URL or email extractions, etc. The $R_2O$ language provides means for specifying declaratively such selections and transformations.

### 3.2 Knowledge Parser: Knowledge Extraction from Documents from the Official Journal Web Sites

One of the objectives of the Fund Finder application is to allow content integration about funding oportunities coming from different legacy sources. This is particularly useful in the domain of fund finding because sometimes the information related to a specific funding opportunity is not complete or is spread over several Web sites. For instance, the CIDEM's database does not contain information about the documentation that a candidate needs to provide to apply for a specific funding opportunity. However, it contains the number and date of the official publication from which the information about the funding opportunity was taken, so that it is easy to locate it in the corresponding journal Web site by building automatically the URL of the on-line version of the journal or performing a search over the search facilities of the on-line journal.

Once the relevant document and the specific piece of text describing the funding opportunity are found, the relevant information has to be extracted. In these documents this information is usually available in natural language or in a semi-structured form (normally in the form of a bullet list where the types of necessary documentation are listed). $R_2O$ and ODEMapster cannot be used for this purpose, and hence we used iSOCO's[5] Knowledge Parser® [4], an automatic annotation system able to parse unstructured or semi-structured content, extract knowledge from it and populate it in an ontology.

### 3.3 Semantic Content Integration

Figure 3 shows how the task of integrating the information coming from the different sources is performed. We can see how two mappings (represented by arrows named Mapping1 and Mapping2) have been defined between columns in the database and properties in the ontology in an $R_2O$ mapping description document. The

[5] Intelligent Software Components (http://www.isoco.com/).

ODEMapster processor generates automatically instances for the ontology according to these mapping definitions with the information that it extracts from the database.

The official publication column in the database needs more complex processing. This record is filled in with semi-structured natural language text. This column contains all the elements needed to build the URL of the on-line version of the publication. These elements are extracted using regular expressions, which are also specified in the mapping description document. The generated URL is provided as an input to Knowledge Parser®, which retrieves this on-line resource from the Web and performs the information extraction process to generate instances with complementary information to that obtained from the database. Both instances (named Instance 85 in the diagram) are finally assembled in the resulting knowledge base.

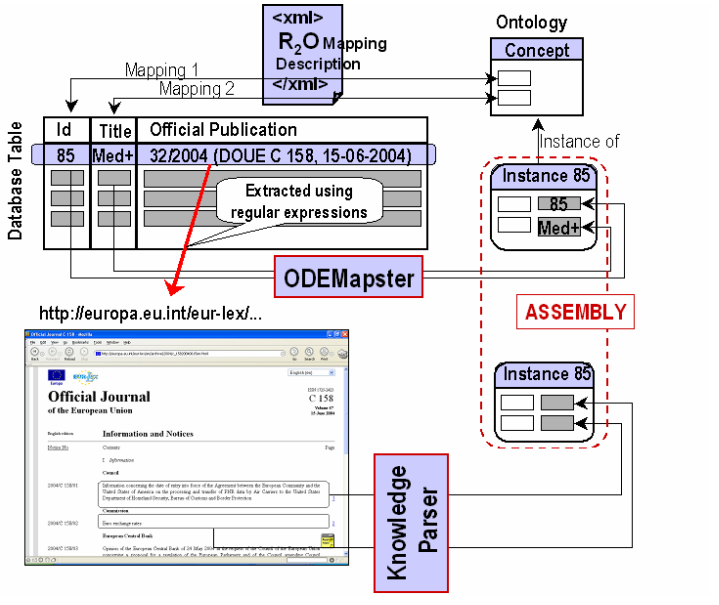


**Fig. 3.** Information integration in the Fund Finder application

## 4 Semantic Publishing and Navigation

We explored two approaches for making the Fund Finder application publicly available, that is, for publishing the Semantic Web portal: batch and on-demand approaches. The selection of a specific approach has an important influence on the way that the knowledge extraction tools (ODEMapster and Knowledge Parser) are used. Figure 4 shows a schematic view of these two approaches, together with some screenshots of their user interfaces.

The batch approach is intended for massive batch semantic content generation and is especially useful when data does not change too often (as it is the case for this application). It is based on a three-step process. First, the content is extracted from the database by the ODEMapster processor and from the official journal documents by

the Knowledge Parser, and is represented in RDF. Then this content is imported into the WebODE workbench using WebODE import services. Finally the content is presented to the user using the ODESeW portal [5], which provides functionalities for semantic-based navigation and querying, different access control rights for different users, personalisation, etc.

The on-demand approach is focused on query processing and is more adequate when data changes frequently. It provides a lightweight presentation layer on top of a simple semantic query engine. The transformations are made on demand, based on the mapping description documents and configuration files needed by the ODEMapster processor and by Knowledge Parser.

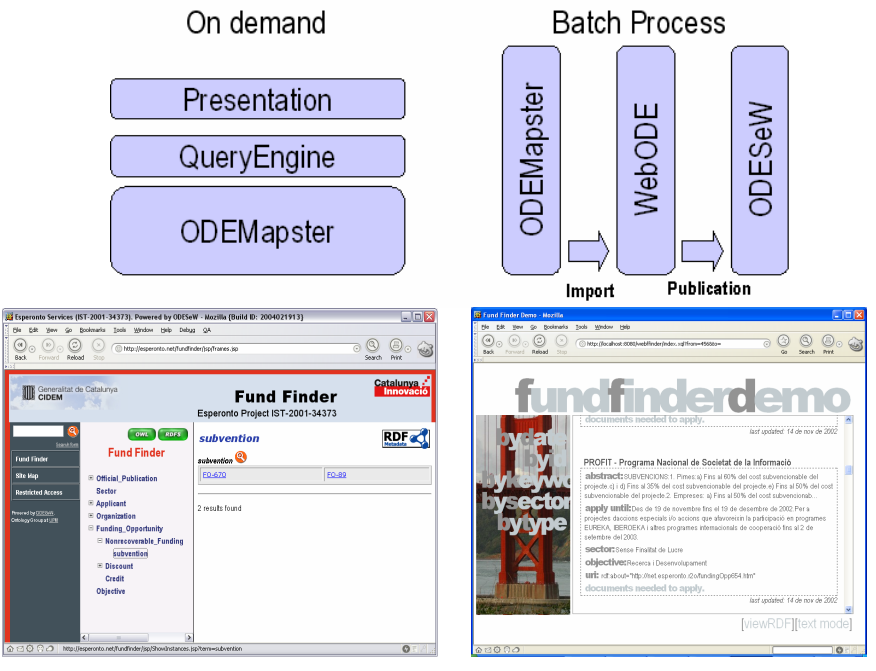


**Fig. 4.** Two alternatives for deploying semantic portals. From left to right, the one generated by ODESeW and the web interfaces that use the semantic query engine.

## 5 Conclusions and Future Work

In this paper we have presented the Fund Finder application, which shows how a set of legacy databases and documents can be upgraded to the Semantic Web with some of the tools developed in the context of the project Esperonto, providing added value by integrating information from different heterogeneous sources, by allowing to perform additional types of queries that cannot be performed with the current application in place, and by allowing another type of navigation that was not foreseen with the current state of affairs.

This application is currently in the evaluation phase inside CIDEM and will be launched in the following year at their Web site, complementing the current application. Both portals are being evaluated and it seems that the early results confirm that the batch mode will be preferred in this application, given the fact that the information sources change only at regular weekly intervals.

Since all the technologies used in the construction of the Fund Finder Semantic Portal are domain independent, they can be easily reused in other domains. The description of another similar application can be found at [4], and other commercial applications are being also developed with this toolset at the time of writing this paper. By providing a toolset for the upgrade of legacy content to the Semantic Web and some hints on how to exploit the upgraded knowledge we strongly believe that we will allow others to implement other similar applications as well, hence fostering the vision of the Semantic Web.

## Acknowledgements

This work has been funded by the European Commission in the context of the project Esperonto Services IST-2001-34373 (http://www.esperonto.net/). We would like to thank Raúl Blanco and Carles Gómara for providing the application requirements and the CIDEM database, as well as Richard Benjamins, Jesús Contreras and Robert Salla for creating the official publication wrappers with Knowledge Parser.

## References

1. Arpírez, JC.; Corcho, O.; Fernández-López, M.; Gómez-Pérez, A. *WebODE in a nutshell.* AI Magazine 24(3):37-48. Fall 2003
2. Barrasa J, Corcho O, Gómez-Pérez A *$R_2O$, an extensible and semantically based database-to-ontology mapping language.* Proceedings of the Second International Workshop on Semantic Web and Databases. Co-located with VLDB 2004 Toronto, Canada, 29-30 August 2004
3. Benjamins VR, Fensel D, Decker S, Gómez-Pérez A. *(KA)2: Building ontologies for the internet: a mid term report.* International Journal of Human-Computer Studies, 51(3):687–712, 1999.
4. Contreras J et al. *A Semantic Portal for the International Affairs Sector.* 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW'04). Springer-Verlag. Lecture Notes in Computer Science (LNCS) 3257:203-215. October 2004.
5. Corcho, O.; Gómez-Pérez, A.; López-Cima, A.; López-García, V.; Suárez-Figueroa, MC. *ODESeW. Automatic Generation of Knowledge Portals for Intranets and Extranets*. Lecture Notes in Computer Science Vol 2870. The Semantic Web - ISWC 2003. Springer-Verlag. pp:802-817. October 2003.
6. Fernández-López, M.; Gómez-Pérez, A.; Pazos-Sierra, A.; Pazos-Sierra, J. 1999. *Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment*. IEEE Intelligent Systems & their applications. January/February PP. 37-46.
7. Staab S, Angele J (2000) AI for the Web - Ontology-based Community Web Portals. 17th National Conference on Artificial Intelligence and 12th Innovative Applications of Artificial Intelligence Conference (AAAI 2000/IAAI 2000), Menlo Park/CA, Cambridge/MA, AAAI Press/MIT Press.
8. Studer R, Benjamins VR, Fensel D (1998). Knowledge engineering: Principles and methods. IEEE Transactions on Data and Knowledge Engineering, 25:161-197.

## Appendix: The $R_2O$ language and the ODEMapster Processor

### The $R_2O$ language

$R_2O$ is a declarative, XML-based language that allows the description of arbitrarily complex mapping expressions between ontology elements (concepts, attributes and relations) and relational elements (relations and attributes). The strength of the $R_2O$ language lies in its expressivity and in its DBMS independence. The elements of the language providing such qualities are **conditions** & **operations** and the **rule-style mapping definition for attributes**.[6]

### Conditions and Operations

Conditions and operations allow the description of *"under which circumstances a database individual (a relational tuple, a database record) can be upgraded to a Semantic Web individual (an instance of the target ontology)"* and *"what kind of transformations are needed to create a Semantic Web individual from a database individual"* respectively. Both are defined in terms of an extendable set of primitives and are identified by their names and the set of named parameters they accept. The values of such parameters can be constant values (**has-value**), variables referring record fields from the database (**has-column**), or the result of the execution of other operations (**has-transform**).

The first $R_2O$ excerpts describe a condition based on the *"match-regexp"* primitive. The condition is verified if the content of column *salaryRange* of table *jobs* matches the regular expression.

```
condition "match-regexp"
    arg-restriction
      on-param "string"
      has-column jobs.salaryRange
    arg-restriction
      on-param "regexp"
      has-value ([:digit:]*)-([:digit:]*)
```

The second fragment describes an operation based on the *"concat"* primitive. The operation concatenates two constant strings with the content of column *id* of table *jobs*.

```
operation "concat"
    arg-restriction
      on-param "string1"
      has-value "http://net.testing.r2o/job-"
    arg-restriction
      on-param "string2"
      has-transform
        operation "concat"
          arg-restriction
            on-param "string1"
            has-column jobs.id
          arg-restriction
            on-param "string2"
            has-column jobtypes.code
```

[6] A complete description of the $R_2O$ Language is available in [2].

Other primitives defined in the first version of the language are: *plus, minus, multiply, divide, apply-regexp, in-keyword, hi-tan, lo-than, equals, hieq-than, loeq-than, etc.*

### Attribute Mapping Definitions

Mapping definitions for attributes are defined as sets of *if-then* rules that allow the conditional generation of attribute values as well as multivaluation. The structure of an attribute mapping definition is described by the following example. The value of the ontology attribute *type* is calculated based on the application of the set of rules (**selector**): If the condition part (**applies-if**) is verified, then the action part (**after-transform**) is executed to generate a value.

```
attributemap-def "http://net.testing.r2o/jobs#type"
selector
  applies-if
    condition […condition desc 1…]
  aftertransform
    operation […transformation desc 1…]
selector
  applies-if
  aftertransform …
```

### The ODEMapster Processor

The ODEMapster processor generates Semantic Web instances from relational instances based on the mapping description expressed in an $R_2O$ document. ODEMapster offers two modes of execution (see figure 5): **Query driven upgrade** (on-the-fly query translation) and **massive upgrade batch process** that generates all possible Semantic Web individuals from the data repository.

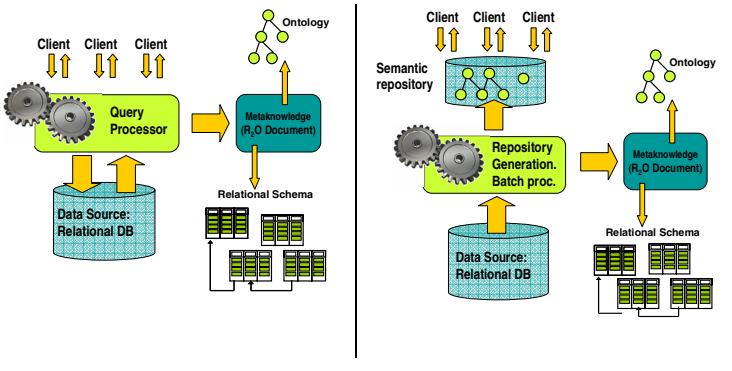


**Fig. 5.** ODEMapster execution modes

The operations of ODEMapster are not limited by the expressivity of the DBMS. The set of primitives can be extended with delegable or non delegable primitive conditions and operations. The processor will delegate the execution of certain actions to the DBMS and execute the rest itself (post processing). The main steps of its executions are: Query & $R_2O$ parsing, SQL generation, SGBD execution result grouping and finally post-processing.