eGoverment interoperability on a semantically driven world

Santos Vicente, María Pérez, Xavier García, Ana Gimeno and Javier Naval
GFI Informática. Serrano Galvache, 56 28033 Madrid
00 34 91 383 63 20
terregov@gfi-info.com

Abstract: In a context of different administrations procedures integration and distributed responsibilities, the capacity of interaction between different agencies regardless of in home nomenclature or language would be a must. Semantic Interoperability Technologies will allow this kind of multicultural environments by the fact of standardising the concepts under a business area and after that, providing definitions based on these concepts for the different procedures provided for all the entities on the system. Once all the procedures on a system are defined according to these standardised concepts, automatic search based on concepts would be a take off point for enabling non supervised information systems interaction and automatic service discovery.

On this article we will put the focus on providing an extended overview about semantic interoperability technologies from the relying concepts to practical experiences which are going on. After providing the overall technological picture, the focus will be granted to semantically enabled web services as one of the most promising approaches to allow the final deployment of these technologies as key enablers of agencies integration. Finally a pragmatic overview will be provided through the presentation of typical use cases and the description of practical experiences coming both from the research and enterprise world. Closing the article, a conclusions section would be provided showing current advantages and weaknesses of semantic technologies.

1 Introduction

Public services within the European countries are demanding increasingly more interconnections among them nowadays. Integration and distributed responsibilities of the agencies involved in the government administrations are required very intensively.
The variety of environments to be interconnected and different types and standards of the information make the integration of heterogeneous administration procedures a complex task. Besides, automatic search processes would be a key feature to allow sharing the common knowledge and services among the interconnected communities.
Semantic Technologies provide a solution to implement the described environments. In this paper, we will focus on providing an overview of interoperability applied to semantic technologies and some practical experiences implemented within European projects.

In the context of public services, the tendency is to move towards a more distributed and interconnected process environment. The new term **eGovernment,** short for electronic government, includes this way of proceeding and implementing public services.
eGovernment uses the information and communication technology in public administrations combined with organisational changes and new skills in order to improve public services and democratic processes and strengthen support to public policies. The potential of eGovernment goes far beyond the early achievements of online public services[1].

The reorganization, modernization and reinvention of government means inter-linking the existing systems to benefit from the exchange of information. Different departments and remote institutions involved will be able to work in a transparent manner.

The standardization of the information and services are represented based on concepts which are used by searches to discover services automatically.
This approach is known as **interoperability** among the systems in an eGovernment environment. It also concerns organisational issues, such as co-ordinating processes that span not only intra-organisational boundaries, but also co-operate with partner organisations that may well have different internal organisation and operations.
Interoperability has been carried out within the European Union by the development of the frameworks and practical experiences applied to public services. These frameworks require data import / export procedures which respect both the integrity of the data and the structure of the exchanged documents, as well as secure procedures such us identity proof, authenticity proof and certification’s procedure. Security is highlighted as an important issue to take into account when deploying these frameworks.

The eGovernment solutions are widely supported by **Web Services** as a way for agencies, other governments, businesses and citizens to make queries and discover the interoperable information available in their systems.
A Web service is implemented as an application that exposes a function accessible using Web technology that adheres to Web services standards. This is a significant feature since Web services are developed for and deployed onto any platform using any programming language. In practical terms, Web services can announce themselves across the Internet and expose their functionality to other applications.

There is a strong momentum to bring Web services technology into the mainstream of network computing. They are seen as fuelling the next major wave of e-business growth and process efficiencies.
Web services appear to become more widely adopted in coming years, allowing much broader integration. Increasingly, developers will require automated systems for service discovery, enabling further Web services interaction with even less human effort.

However, the traditional web services solutions encounter some problems such as information overload and poor content aggregation. The fundamental roots to these problems are the lack of **semantic definitions** in individual systems, the lack of semantic integration between data, and the lack of semantic interoperability across disparate systems.
Semantic technologies appear in this context to extend beyond the capabilities of the current Web and existing information technologies, enabling more effective collaborations and interconnection among systems. eGovernment benefits from these approach when interconnecting the agencies and discover their services in a transparent manner. Service requests are responded by search processes, conceptual frameworks and performing intelligent reasoning.

2 Semantic Technologies

Semantic technologies provide a very innovative implementation for eGovernment solutions to solve the necessities within public services to exchange and interoperate among the different agencies regardless their location or nature of their information. Semantics add the key feature to standardise the information as concepts and to enable the automatic service discovery. Information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.
The difficulty in ensuring flexible discovery and service initiation, as well as operational use of information exchanged with Web services, has led to incorporate semantic technologies to the traditional web.
The **Semantic Web** is an extension of the current World Wide Web, not a separate set of new and distinct websites. It builds on the current Web adding further capabilities by defining machine-processable data and relationship standards along with richer semantic associations.
Building a Semantic Web is the next phase of the current web and adheres the semantic benefits to the already broadly used web services.
Figure1 bellow shows a conceptual stack for the Semantic Web, illustrating how semantic technologies can be added to extend the capabilities of the current web.

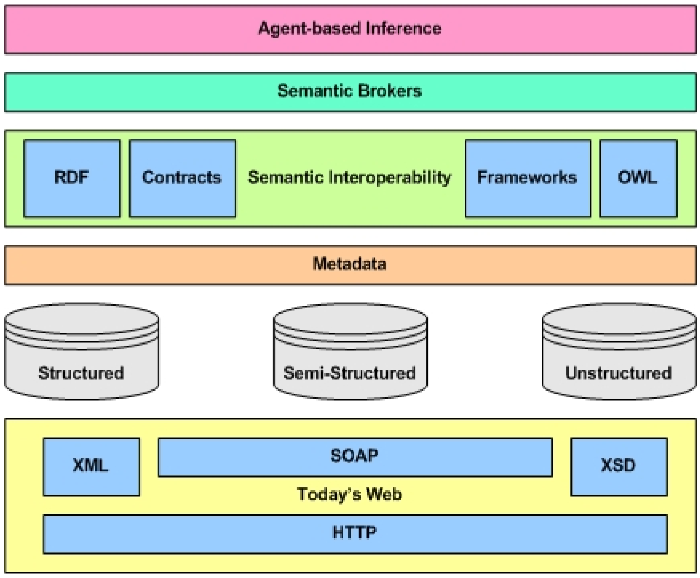

Figure 1: Semantic Web Conceptual Stack

The implementation of the semantic web is supported by semantic technologies which are capable of representing, acquiring and utilizing the knowledge.
When representing knowledge in semantic web, an **ontology** is used to describe and represent the interrelationships of the information. An ontology adds semantic features to the traditional information available in the web.
Moving from traditional web to semantic web is supported by a number of available tools to automate the process of adding knowledge and incorporating semantic descriptors from the defined ontology to the existing web services. Additional information can also be inferred by agents using the semantics defined.

In Feb 2004, The World Wide Web Consortium released the Resource Description Framework (RDF) and the OWL Web Ontology Language (OWL) as W3C Recommendations. RDF is used to represent information and to exchange knowledge in the Web. OWL is used to publish and share sets of terms called ontologies, supporting advanced Web search, software agents and knowledge management.

RDF, Resource Description Framework, is used to describe any Internet resource such as a Web site and its content. An RDF description (referred as metadata) can include the authors of the resource, date of creation or updating, the organization of the pages on a site, key words for search engine data collection, subject categories, etc
OWL, Web Ontology Language, is a markup language that enables ontology sharing via the Web.
Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge can be reusable.
An ontology is formed by a set of concepts, their attributes and relationship among them, describing knowledge about a specific domain, and being intelligently shared between people, databases and applications.
Following is an example of an ontology which is defined in a XML file with *owl* extension:

```
<?xml version="1.0" ?>
<rdf:RDF
```

```
xmlns="http://osm.cs.byu.edu/CS652s04/ontologies/OWL/carads.owl#"
xmlns:carads="http://osm.cs.byu.edu/CS652s04/ontologies/OWL/carads.owl#"
xml:base="http://osm.cs.byu.edu/CS652s04/ontologies/OWL/carads.owl#"
xmlns:example="http://osm.cs.byu.edu/CS652s04/ontologies/annotatedPages/carSrch1_semweb.html#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
<owl:Ontology rdf:about="">
<rdfs:comment>OWL ontology example </rdfs:comment>
<rdfs:label>Car Advistisement Ontology</rdfs:label>
</owl:Ontology>
<owl:Class rdf:ID="CarAds">
    <rdfs:label xml:lang="en">CarAds</rdfs:label>
    <rdfs:subClassOf>
      <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMake" />
      <owl:minCardinality
      rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
…
   <owl:DatatypeProperty rdf:ID="featureValue">
      <rdfs:domain rdf:resource="#Feature" />
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
   </owl:DatatypeProperty>
   <owl:ObjectProperty rdf:ID="hasMake">
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
      <rdfs:domain rdf:resource="#CarAds" />
      <rdfs:range rdf:resource="#Make" />
   </owl:ObjectProperty>
   <owl:ObjectProperty rdf:ID="featureFor">
      <owl:inverseOf rdf:resource="#hasFeature" />
   </owl:ObjectProperty>
   <owl:DatatypeProperty rdf:ID="carAdsValue">
      <rdfs:domain rdf:resource="#CarAds" />
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
   </owl:DatatypeProperty>
…
   <CarAds rdf:ID="CarAdsIns1">
     <carAdsValue
          rdf:datatype="http://www.w3.org/2001/XMLSchema#string">001</carAdsValue>
    </CarAds>
   </rdf:RDF>
…
```

The ontology expresses the domain knowledge in the form of concepts which are defined by classes (*owl:Class* tags); subclasses which define the hierarchy and relationships of these concepts (*owl:subClassOf* tags); attributes of the concepts which are defined by properties (*owl:DatatypeProperty, owl:ObjectProperty* tags) and instances of the classes, which are the actual information values (*CarAdsIns1* in the example above).

In the context of web services, Web Ontology Language (OWL) led to the definition of another standard to cater for the web services requirements. The standard is called **OWL-S**.
OWL-S supplies a core set of markup language constructs for describing the properties and capabilities of their Web Services in unambiguous, computer-interpretable form. OWL-S markup of Web services will facilitate the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and execution monitoring. Following the layered approach to markup language development, the current version of OWL-S builds on top of W3C's standard OWL.
OWL-S enables declarative advertisements of service properties and capabilities that can be used for automatic service discovery. It can also perform automatic invocation of a web service by a computer program or agent, given only a declarative description of that service.

There is an API that includes the semantics of the arguments to be specified when executing the web services, and the semantics of that is returned in messages when the services succeed or fail.
OWL-S also allows automatic web service composition and interoperation of web services to perform some complex task, given a high-level description of an objective. It provides declarative specifications of the prerequisites and consequences of application of individual services , and a language for describing service compositions and data flow interactions.
The description of a web service in OWL-S consists of the following layers:

– a *Profile*, which tells "what the service does",
– a *Process Model*, which tells "how the service works",
– and a *Grounding*, which tells "how to access the service".

Following the layered approach to markup language development, the current version of OWL-S builds on top of OWL. The following example gives the upper part of the ontology with the concepts of service, profile, process and grounding.

```
<!-- Service -->
<owl:Class rdf:ID="Service">
  <rdfs:label>Service</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
</owl:Class>

<!-- Service Profile -->
<owl:Class rdf:ID="ServiceProfile">
  <rdfs:label>ServiceProfile</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
</owl:Class>

<!-- Service Model -->
<owl:Class rdf:ID="ServiceModel">
  <rdfs:label>ServiceModel</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
</owl:Class>

<!-- Service Grounding -->
<owl:Class rdf:ID="ServiceGrounding">
  <rdfs:label>ServiceGrounding</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
</owl:Class>
```

Each of the classes defined in this top layer are defined in 3 different *.owl* files following the owl standards:

– **Service.owl:** sets all the properties of the service *<owl:ObjectProperty>*
– **Profile.owl:** definition of the Profile class. Details of the service are specified, such as, service profile, service name, contact, text description, contact information, functionality description (hasParameter**,** HasInput**,** hasOutput **,** hasPrecondition**,** hasResult), profile attributes (serviceParameter, serviceCategory) …
– **Process.owl:** process parameter Class with details such as participants, inputs, outputs, preconditions and results.
– **Grounding.owl:** grounding instances and definitions for grounding (protocol and message formats, serialization, transport, and addressing).

3 Applications in eGovernment

Web services are the most chosen implementation in the eGovernment context. Several European projects are in progress providing solutions to implement the administration procedures and interaction with citizens.

The European project **SWAD** (Europe or Semantic Web Advances Development in Europe) aims to highlight practical examples of where real value can be added to the Web through the Semantic Web. SWAD provides real examples of how the Semantic Web can address problems in areas such as: sitemaps, news channel syndication ; thesauri, classification, topic maps; calendaring, scheduling, collaboration; annotations, quality ratings, shared bookmarks; Dublin Core for simple resource discovery; Web service description and discovery; trust and rights management ; and how to mix all these together.

Following is a description of some of the projects being developed in this context:

- Integrate an Open System City Platform: it provides interactive city-wide online applications and services for users that make all aspects of what is “going on” in the city available to everyone ([11] *Intelcities*).
- Enable local governments to manage their policies in a transparent and trustable way through an actual evaluation of services performance and of citizens satisfaction and expectations ([12] *Qualeg*).
- Ontology-based methodology/platform for facilitating consistent composition, reconfiguration and evolution of eGovernment services ([13] *Ontogov*).
- Open, secure, interoperable and cost effective eGovernment platform for municipal eGovernment applications ([14] *Emayor*).

3.1 TERREGOV

The TERREGOV project is an European Union funded project which aim is to enable local governments to deliver online services, specially in the Social Care environment, in a straightforward and transparent manner regardless of the administrations actually involved in providing those services.

Semantic enrichment of web services is the philosophy developed by TERREGOV in which web services and plain text documents are described using an ontology. Such an ontology provides a language-independent mechanism to support the automatic discovery of an answer to a citizen’s request regardless of administrations or organization. A constraint is that all information exchanged has to be achieved through the translation of data into the unique ontology for TERREGOV and always based on semantic knowledge of information in accordance with rules for interconnectivity and dynamic discovery.

Access to web services could be transparent but with restrictions because of the different legislative constraints and authorization limits in the different administrations involved.

TERREGOV´s **requirements** include the use of standards for implementing web services; use of ontologies for structuring knowledge, allowing to describe web services unambiguously, implementing human-machine interfaces, indexing and retrieving information; use of natural language processing for automating partially the ontology developments and improving the human-machine dialogs.

Overall, the goal of the project is to deliver services in a transparent manner.

The TERREGOV solution provides ontologies used for organizing the relevant knowledge, and indirectly driving the business processes. They will be useful in different tasks, namely: (i) structuring knowledge for knowledge management, leading to the enrichment of web services meta-data with semantic descriptors; (ii) driving Human-Machine dialogs; (iii) semantic indexing/retrieval of text and documents.

The implementation of the project takes place in different European pilots in which administrations in the context of social care are involved.

The most expected **use case** of TERREGOV is that a civil servant, point of contact between citizen and public administration, is asked about a problem. In order to solve it and to report the solution to the citizen, the civil servant, through the use of TERREGOV solution, can collaborate with other civil servants and experts to get hints; get access to the specific knowledge base, and search for information; discover the best service for the specific citizen's case and invoke the execution of the selected service and monitor its execution.

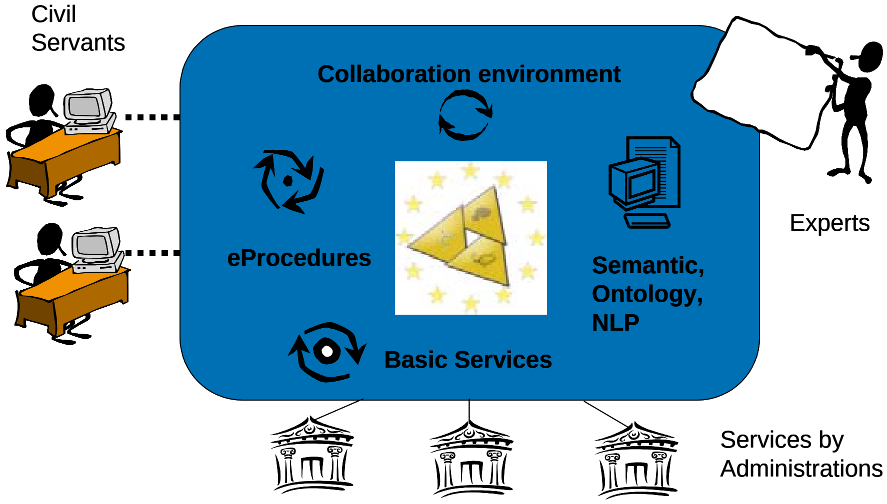

Figure 2: Terregov project: environment diagram

The citizen can call on a range of services helped by an administrative agent or civil servant to make requests to public services. With search tools based on a semantic approach and natural language, the civil servant or the citizen will be able to call e-procedures at a local, county, regional or national level.

Each administration is responsible for defining the web services it wants to offer. The functionalities include first, the dynamic discovery of the Web Service using semantic description and searching. Second, an interface that allows electronic registration and entry of citizen's data. This interface will have complete tools like an electronic agenda for notes and appointments, an email connection service with other civil servants for information exchange, automatic printing of documents or automatic generation of letters for citizens and connection with database systems. Third, the storage of information.

A registry of citizens' personal data and their completed requests are stored in databases. Then the TERREGOV web service will automatically offer a specialized referent based on a citizen's own criteria and will update the citizen's record. The file is stored in the database and the file is sent electronically to a buffer zone. Storing the queries made to the system could help TERREGOV in learning about the efficiency of requests, optimum indexing of web services and strengths/weaknesses of connection among administrations. The answer is generated through the steps, defined in workflows, that are needed to get a dynamic discovery of services.

TERREGOV adopts the principles of a service oriented **architecture** (SOA) based on interoperable components with dynamic support for finding services. The idea is strengthened by the fact that information, services and administrations are spread over several information systems. The architecture contains a set of collaborative tools for eGovernment web services semantically enriched.

The solution is built on a multi-layer approach where citizens from any location and using different languages can access the services. This view is in line with the *Clearing House*

approach, where a global server providing different applications over a common back-office is used to solve interoperability and semantic problems.
The figure bellow illustrates the functional architecture used in TERREGOV.

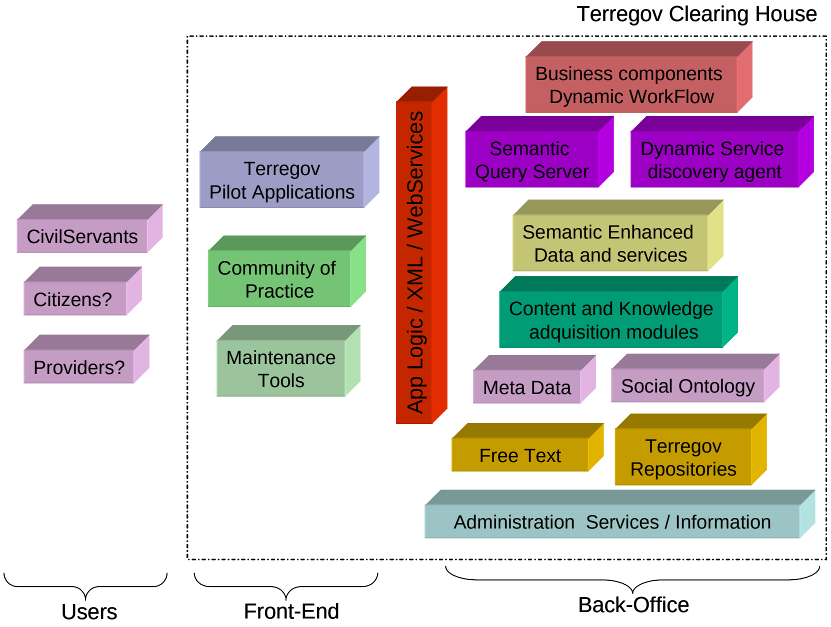

Figure 3: Terregov´s functional architecture

The *Frond End* layer is composed of different applications demanded from the system. The platform is potentially extensible to enable later integration of new applications.
A set of pilot applications and a community of practice are defined as well as an additional administration interface.
The *Back Office* layer is the internal group of subjacent technical modules. It integrates all the infrastructure required in order to operate on top of different organizations offering different services, information structures and languages.
The technological modules deal with the interoperability issues. Intelligent agents implement mechanisms to dynamically select web services to be invoked according to performance indicators and to integrate semantic negotiation technology.
Besides, there is a Web Services Semantic Binder module to enable enriching web services in the registry with semantic descriptions from ontologies. These can come from various sources and fusion tools will merge them to make them compatible.

Finally, it is worth highligting the fact that TERREGOV is potentially an important step towards the next generation of Internet-based interoperables services enriched with semantic descriptions. Up to now, most services offered no semantic description of their input, outputs, or of how to use them. The use of ontologies to describe various aspects of the services will permit to make explicit these descriptions, and allow for computer-based interoperability within eGovernment environments.

4 Conclusions

From this study one can infer that web services and ontologies have very little in common. Indeed, approaches and formalisms are very different. Web services are developed within the web technology and are grounded on a firm XML basis. Ontologies were developed in the field of artificial intelligence and use more esoteric formalisms. However, ontologies are beginning to migrate towards the web environment. In addition, web services lack a semantic

layer despite the existence of meta-data, which makes web services and ontologies complementary. Efforts are being produced to match semantic approaches with web services techniques. It can be also concluded the existence of a number of available open source tools in each domain. Since the field of semantic description and enrichment of Web Services is rather recent, knowledge about this approach is not yet widespread, and there are very few applications already deployed that use such semantic information.
Semantic solutions in the eGovernment domain are being implemented in slow progress evolving along with the advance in semantic technologies.

Practical experiences show difficulties to standardise the structure of the public administrations involved in the eGovernment interoperability. Efforts should be put during the first stages of the implementation of the solutions to define the structures as abstract as possible. There are also legislative constraints and authorization barriers that should not be discarded to be able to access the web services in a transparent manner.

REFERENCES

[1] Europa , portal site of the European Union <http://www.europa.eu.int/>, Jan 2005.
[2] Terregov, Impact of eGovernment on Territorial Government Services, *Terregov D1.1 Technological State of Art v1*, < http://terregov.eupm.net/ >, Jan 2005.
[3] Vasilis Peristeras, Prof. Konstantinos Tarabanis, *Towards an Information Architecture for Public Administration*, 2001.
[4] Gómez-Pérez, Fernández-López Corcho. *Ontological Engineering*. Springer 2004.
[5] Fensel D., C. Bussler, and A. Maedche. *Semantic Web Enabled Web Services*. In *ISWC 2002*. Vol. pp.1 –2:LNCS 2342, Springer-Verlag, Berlin Heidelberg. 2002.
[6] W3C, *DAML+OIL Reference Description* <http://www.w3.org/TR/daml+oil-reference>, Nov 2001.
[7] DARPA, *DARPA Agent Markup Language Web Services (DAML-S) Version 0.7 Specification*, 2002, <http://www.daml.org/services/daml-s/0.7/>, Nov 2002.
[8] SICoP, *Introducing Semantic Technologies and the Vision of the Semantic Web*, Sept. 2004 <http://colab.cim3.net/file/work/SICoP/WhitePaper>.
[9] Web Services Journal, *Semantic Discovery for Web Services*, April 2003. <http://www.sys-con.com/webservices/article.cfm?id=507>.
[10] Semwebcentral, Open Source development web site for the Semantic Web, <http://semwebcentral.org/>, Jan 2005.
[11] Intelcitiesproject , Research and technological development project for e-government services. <http://www.intelcitiesproject.com/>, Jan 2005.
[12] Qualeg , European project for Quality of Service and Legitimacy in eGovernment <http://www.qualeg.eupm.net/my_spip/index.php>, Jan 2005.
[13] Ontogov, European project for Ontology-enabled e-Gov Service Configuration http://www.ontogov.com, , Jan 2005.
[15] Web Services Journal, *Semantic Discovery for Web Services* <http://www.sys-con.com/webservices/articleprint.cfm?id=507>, Jan 2005.
[16] Ralph Hodgson, *The Potential Of Semantic Technologies for e-Government*, eGov Conference 2003 <http://www.coolheads.com/egov/opensource/topicmap/s167/abstract.html>
[17] Terregov, Impact of eGovernment on Territorial Government Services, *D4.2&4.3 Definition of TERREGOV Prototype*, < http://terregov.eupm.net/ >, Jan 2005.
[18] Ontology example <http://osm.cs.byu.edu/CS652s04/ontologies/OWL/carads.owl>, Jan 2005