A Framework of Public Administration Domain Data Interoperability Using Semantic Web Technology

Sotirios K. Goudos, Vassileios Peristeras, Konstantinos Tarabanis
Informatics and Telematics Institute, Center for Research and Technology Hellas, 1st km Thermi-Panorama Road, Thermi, 57001, Thessaloniki, Greece, {sgoudos, per, kat}@uom.gr

Abstract. This paper addresses the problem of supplying the citizen with the correct information from the public administration in a quick and easy way. Getting an operation license in Greece for any kind of an enterprise is a very complex task. The OWL web ontology language is used in order to create a suitable ontology for the problem described above. The ontology is used with the help of a reasoner through a web interface to extract the correct information. The results give the documents and the procedure required for getting operation license for various enterprise types. That way the whole procedure is simplified omitting steps of searching in legislation. The proposed framework can be applied in similar cases in the public administration domain. Finally the advantages of semantic web technology are discussed.

1 Introduction

The eEurope Action Plan has established quantitative and qualitative indicators for benchmarking the progress of e-Government in member countries. In establishing indicators for e-Government, the approach taken is to focus on the demand side, i.e. the bottom-up reality of citizens’ and businesses’ contacts with government. For the e-Government indicators, the following maturity model is used:

- Stage 1 Information: online info about public services,
- Stage 2 Interaction: downloading of forms,
- Stage 3 Two-way interaction: processing of forms, incl. authentication,
- Stage 4 Transaction: case handling; decision and delivery (payment).

Information services available on-line from the public administration is the first stage in the e-Europe model maturity for e-government. This stage has drawn a great amount of effort from many EU countries in order to satisfy the demand for faster information available. The implementation of an information system that will serve that demand is not always an easy task. This is due to the inherited difficulties that exist in public administration domain. There are many complicated services with numerous executional paths, depending on the type of process. In many cases in the EU countries various legislation exists for the same goal that complicates the search effort for the average citizen. It is very difficult in several
cases for a citizen to be able to find through the legislation all the correct information. This information entails all the formal documents and the procedure that is required to follow for a certain task.

Our motivation was to find a flexible and scalable framework and to implement an information system for such complex cases. To solve this problem, we have chosen the semantic web technologies, and propose an infrastructure that could simplify the procedure.

The following sections are organized as follows: A technology overview is presented in section 2 and the business case is given in section 3. The proposed system architecture is in section 4 which is divided in the ontology and reasoner section. A use case example is given in section 5. Finally section 6 contains the conclusion and future work.

2 The State-of-the-art

Several information technologies exist for the creation of web-based e-government applications. Data representation in a relational model is very popular and it is widely used. Creating dynamic web pages linked to an external relational database is a very common task. A more recent technology is the semantic web. The use of Semantic Web [1] and Semantic Web Services [10] technologies to enable the interoperability of systems and applications is gaining momentum worldwide.

The state-of-the-art technology in a web environment is adding semantic meaning to web resources. Currently these resources are usually only human understandable: the mark-up (HTML) only provides information for textual and graphical information intended for human consumption. Semantic Web aims for machine understandable information that can be processed and shared by both computers and humans. Tim Berners-Lee [1] provides the definition of the Semantic Web as “an extension of the current one [Web], in which information is given well-defined meaning, better enabling computers and people to work in cooperation.” Of course the semantic technology is not a competitor of the relational model since they apply to different types of applications for different reasons but still there are areas of overlapping.

It is not an easy task to design the relational data model to represent a complex scenario. It is even more complicated to create the appropriate SQL queries to extract the correct information. In simpler cases the relational model can be used. But in complicated cases an ontology model provides more flexibility and robustness in design and implementation. The advantage of semantic technology over the relational model lies on the fact of creating machine-readable data capable of modelling complex cases. The same information can be shared not only among humans but also among clever agents in the web. The scalability is also an advantage for the semantic technology. More details about the question answering capabilities of the semantic web can be found in [7].

Data representation in a semantic web environment is given in four layers as shown in Fig. 1 [4]. These layers are XML, RDF (Resource Description Framework), Ontology (OWL), and Logic. OWL is an ontology language for the Semantic Web, developed by the World Wide Web Consortium (W3C) Web
Ontology Working Group [2]. In OWL, ontology is a set of definitions of classes and properties. OWL has the ability of applying constraints on the way those classes and properties can be employed.

There are three sublanguages of OWL: OWL Lite, OWL DL and OWL Full. OWL Lite supports those users primarily needing a classification hierarchy and simple constraints. OWL DL (Description Logic) supports those users who want the maximum expressiveness while retaining computational completeness. OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL (McGuinness (2004). Description Logics (DLs) is a family of knowledge representation (KR) formalisms that represent the knowledge of an application domain by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain.

OWL Full extends OWL DL by adding the syntactic freedom of RDF with no computational guarantees. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full [3]. Currently there are no reasoners available for OWL Full. Therefore OWL DL was the sublanguage selected for the creation of our ontology.

3 Description of the Business Case

Following the above-mentioned line of work, we have used a quite complex service as provided in Greek, namely the issuance of operation licence to companies, as a test-bed for semantic web technology. This paper addresses this issue by proving a solution using this technology.

In Greece getting an operations licence for any kind of enterprise is a complex task. The Prefecture issues both the registration and the operations license. First it requires the applicant to enter a process for obtaining a registration license. After the registration license task is completed the operations license must be obtained. It involves submitting the appropriate documents from a number of public services and authorities. These documents are submitted to distinct departments of the Prefecture, which issues a certificate for every type of document. These certificates together with the registration license are again submitted to another department of the Prefecture. Finally the Prefecture issues the operations licence.
Every enterprise in Greece should submit documents needed based on three major factors that influence the whole enterprise. These are; the number of employees that the enterprise will employ, the location of the enterprise (urban, suburban or rural) and the power in KW needed for the operation of the enterprise. Every one of these attributes has three different possible states. Each state requires a document from a different public service. For example if the number of employees is less than 100 people then only a legally signed statement is required. If the number of employees is between 100 and 1000 then a certificate from the public labour organization (OAED) is required. Finally if the number of employees exceeds 1000 people then a certificate from the Ministry of Industry is required. If the enterprise is situated in an urban area then a certificate from the local municipality is required. If the enterprise is in a suburban area then a certificate from the prefecture must be obtained and if it is in a rural area then a document from the regional authority is obligatory. The operation power needed plays also an important role. So when the power is less than 12KW then a technical report written and signed by an engineer must be submitted. When the power is between 12-50KW then an approval from the Technical Chamber of Greece is required. In case of a power larger than 50KW then the approval of the Ministry of Power must be obtained.

Another issue in Greek legislation is the proximity of an enterprise to specific sites. For example the proximity to archaeological sites requires an approval form the Archaeological Service. The proximity to civil airports requires an approval form the Greek Civil Aviation. If the enterprise is close to a forest the approval of the Ministry of Agriculture is also required. Finally if the enterprise resides in a coastal region then the approval of the Greek Tourist Organisation is required. When all the required documents are gathered the Prefecture issues a Certificate for every case. When all the required certificates are obtained the operations license is issued.

Therefore anyone who wants to start a new enterprise must search in various and complex legislation in order to find the correct documents required. The above scenario has 108 possible combinations. An efficient way to implement electronically the above cases is using semantic web technology, as it will be presented in the next section.

4 System Description

Fig. 2 shows the system architecture. It consists of a web server, a reasoner and an OWL file which is used as the knowledge base. The users access the application through a common Internet browser. The advantage of every web-based front end is that it requires only an Internet browser in order to execute and it can be access from anywhere in the Internet. The system architecture used is server-side therefore the client only shows the form and the results page.
The server uses the data given to invoke the reasoner. The reasoner sends various queries to the knowledge base. The answers returned are parsed to the web server that creates the results web page. The extracted results contain a list of the documents required in order to obtain an operation license. More specifically the web server used was Apache Tomcat, the knowledge base was an OWL file with the ontology definition and the reasoner was JTP (Java Theorem Prover) [5]. The system components are explained in detail in the next sections.

4.1 The Ontology

The business case presented in section 3 can be modeled with a set of OWL classes and properties. The ontology has been created using the Protégé tool with the OWL plug-in [8], [9]. The classes are given schematically in Fig. 3. These consist of four major classes: Document, Enterprise, Certificate and EnterpriseDescriptor. More specifically:

- **Class Document** is the abstract representation of the documents required.
- The **Enterprise** class defines the concept of the enterprise. The various types of enterprises like a bakery store or a mining industry are modeled as subclasses of this class.
- The **Certificate** class represents the concept of the certificate required for each attribute of the enterprise.
- **EnterpriseDescriptor** class is a superclass which represents the properties of the Enterprise.

Individuals in OWL represent objects in the domain that we are interested in. Individuals can also be referred as instances of classes. Class Document has four subclasses LocationDocument, CategoryDocument, PowerDocument and ProximityDocument. The documents required for every case are modeled as individuals of these classes.

- **LocationDocument** class represents the documents required due to the location of the Enterprise. For example the document required in case of an Enterprise located in an urban area is an approval from the local municipality.
CategoryDocument class represents the documents required due to the employee number of an enterprise.
PowerDocument class is an abstract representation of the documents required due to the operating electrical power of the enterprise.
ProximityDocument represents the various documents required in case of proximity to certain sites. For example if the enterprise is close to a forest then an approval from the ministry of agriculture is required.

The various features that describe an Enterprise are given as subclasses of the EnterpriseDescriptor Class. The four subclasses are Power, Location, EmployeeNum and SiteProximity. These classes are the abstract representations of the properties that characterize an enterprise according to current legislation. The different states of each subclass again are given with distinct individual.

Power subclass represents the different operating power ranges in KWatts for every enterprise. An enterprise may have operating power less than 12KW, between 12 and 50KW or more than 50KW.
Location subclass represents the concept of the location area of an Enterprise. There are three individuals of Location class Urban, SubUrban and Rural.
EmployeeNum class represents the category of the employee number of the enterprise. Individuals of these classes are less than 100 employees, between 100 and 1000 and more than 1000 employees.
SiteProximity class is the abstract representation of the proximity of an enterprise to a specific site. For example one individual of this class is Forest, which represents the concept of a forest located close to the Enterprise.

The class Certificate has also four subclasses. Each subclass has other subclasses, which represent the specific certificates required for each EnterpriseDescriptor subclass.

In OWL Properties represent relationships between two individuals or between an individual and a data value. Two types of properties exist, Object properties and Datatype properties. The Object properties link an individual to an individual and the datatype properties link individuals to data values such as an integer, a string or a Boolean.

A number of object properties are also created for the above classes. The object properties for describing the different attributes of an Enterprise are belongsToCategory, LocatedIn, hasPower, and locatedNear. The first three are functional properties and their domain is Certificate class, while their range corresponds to the respective EnterpriseDescriptor subclass. The definition of property belongsToCategory is given in Fig. 4.
Fig. 3. The classes that appear in the ontology

```
ObjectProperty {belongsToCategory Functional
    domain (Certificate)
    range (EmployeeNum)}
```

Fig. 4. The definition of the functional property belongsToCategory in OWL Abstract Syntax

The property locatedNear is defined as an object property (which implies multiple values) and its range is defined the in the SiteProximity class. The object property hasLicense has domain the Enterprise class and range the Certificate class. This property represents the certificates needed for the final operations license. Each Enterprise as stated earlier requires at least three different certificates. The property definition is given in Fig. 5.
Another object property is hasDocument, which is defined with domain the Certificate class and range the Document class. This property represents the documents required in order to obtain a certificate of each category. Each subclass of the Enterprise class is defined with the appropriate hasLicense property values.

In OWL, restrictions are used to restrict the individuals that belong to a class. There are three types of restrictions: quantifier restrictions, cardinality restrictions and hasValue restrictions. The restrictions are used in class definitions. For example the MiningIndustry subclass definition is given below in Fig. 6. One may notice that the definition of the MiningIndustry class follows both quantifier restrictions and hasValue restrictions. The definition shown below means that all the values in hasLicence property (quantifier restriction) come only from those individuals that have value (hasValue restriction) in the property LocatedIn the individual Rural. The same quantifier restriction applies also for the belongsToCategory property. The hasLicense property can be linked only to individuals that their belongsToCategory property has the value MoreThan1000Emp. A certificate definition example is given in Fig. 7.

One may notice that the restriction is applied on hasDocument property, which holds the value of the document required for this certificate. The condition defined as sufficient and necessary for class RuralAreaCertificate (here shown with the owl:equivalentClass tag) is that the value of the property locatedIn has the value Rural.

Fig. 5. The definition of the property hasLicense in OWL Abstract Syntax

```
ObjectProperty(hasLicense domain(Enterprise)
               range(Certificate))
```

Fig. 6. The definition of the Class MiningIndustry in OWL Abstract Syntax

```
Class(MiningIndustry partial restrictions(hasLicense allValuesFrom
                                      (restriction(belongstoCategory value(MoreThan1000Emp)))
                                      LocatedIn(rural)(Enterprise)))
```

```
Class(RuralAreaCertificate complete restriction(locatedIn value(Rural))
       locationCertificate)
```

```
owl:equivalentClass(RuralAreaCertificate restriction(hasDocument value(Rural)(Enterprise)))
```

Fig. 7. The definition of the Class RuralAreaCertificate in OWL Abstract Syntax

4.2 The Reasoner

The reasoner used was JTP (Java Theorem Prover) [5]. JTP supports OWL DL knowledge bases, and uses KIF format for queries and answers. When the user
finally presses the submit button, the web server invokes JTP by sending KIF formatted queries and assertions. First a new individual of the Enterprise class is created on the fly. The properties set by the user with the user interface are entered as assertions to the reasoner. Finally, queries are performed to ask the reasoner for the desired information, which is the list of the specific documents to be submitted by the user. JTP uses the OWL knowledge base and returns finally the extracted results, which include the documents required for every type of a certificate. The system uses JSP (Java Server Page) and Java servlets in order to create dynamically the results in HTML format.

5 Use Case

When the user enters the first page of the application a form appears. The user uses the form to fill in the enterprise details. For a selected enterprise some details are set automatically through the semantic logic. For example if the user selects a Drugstore then the property belongsToCategory is set to LessThan100 employees as it comes from the definition of the Drugstore class. The exact location of a Drugstore however cannot be set from the definition so the user must select it. The proximity to different types of sites is also an option that the user must set or not. When the selection process is over then the user presses the submit button and sends the selected data to the server. The user interface from a demo page is given in Fig 8.

![Fig. 8. a. Application main page; b Sample results page](image)

Let us assume that the user selects a drug store as the Enterprise type. The user also selects that the drug store is located in a suburban area and that is located close to a coast and an archaeological site. The power category is set to less than 12KW and the employee number to less than 100 due to the class definition of the drug store. The results obtained are a complete list of all the documents needed, more specifically in our case these are: a Prefecture approval document, a technical report signed by a civil engineer, a legally signed document, an approval document obtained from the Greek Tourist Organization and an approval document obtained...
from the Greek Archaeological Service. So the user has all the information needed in a quick and easy way. The results from the demo implementation for this case are shown in Fig 8.

6 Conclusions and Future Work

A semantic web application to a common PA service provision in Greece has been presented. The proposed framework and application can help in simplifying the searching procedure for anyone involved in the establishing of a new enterprise. The interoperability between different PA services can be achieved using semantic web technology. The web ontology language OWL approved by W3C has the potential to be applied in e-Government complex cases. In these cases, OWL can increase performance and decrease development time due to its flexibility and scalability. Our future work will include the development of the provision of a complete semantic web service for a common public administration case in Greece.

References