Journal of Web Engineering, Vol.8, No.3 (2009) 245-267

AN ONTOLOGICAL REPRESENTATION OF PUBLIC SERVICES: MODELS, TECHNOLOGIES AND USE CASES

VASSILIOS PERISTERAS

National University of Ireland Galway, DERI, IDA Business Park, Dangan, Galway, Ireland

vassilios.peristeras@deri.org

SOTIRIOS K. GOUDOS

Telecommunications Centre, Aristotle University of Thessaloniki, Thessaloniki, GR-54124, Greece
sgoudo@physics.auth.gr

NIKOLAOS LOUTAS

Information Systems Laboratory, University of Macedonia, 54006, Thessaloniki, Greece
nlout@uom.gr

KONSTANTINOS TARABANIS

Information Systems Laboratory, University of Macedonia, 54006, Thessaloniki, Greece
kat@uom.gr

Received April 8, 2008
Revised March 30, 2009

This paper presents our effort to "ontologize" a conceptual public service model in order to express in a formal way domain specific semantics and create a reusable service ontology for eGovernment applications. The conceptual model we have used comes from a broader public administration domain modeling effort, called Governance Enterprise Architecture (GEA). With this as a starting point, we document our experience of using the Web Ontology Language (OWL) for the ontological representation of the model. Moreover, we present a use case and a platform that is based and uses this ontology for the discovery of eGovernment services. These services are discovered by semantically matching citizens' profiles with formally described public services. The proposed domain ontology is reusable and can be exploited by a variety of semantic web applications for eGovernment whenever a formal and standardized model for public services is needed.

Key words: eGovernment, public administration service provision, semantic web applications, OWL, semantic discovery, ontology
Communicated by: D. Lowe & M.C. Norrie

1 Introduction

EGovernment is an attractive field for research organizations and businesses as well. Recently, there is a growing interest on modeling the public administration (PA) domain as among others these models can be then used for applying semantic web technologies [1-3].

Public Administration (PA) has some certain characteristics that differentiate it from the private sector [4, 5]:

- PA is a diverged and distributed environment layered in clearly defined organizational levels (e.g. local, regional, national).
- PA is hierarchically organized and governed by explicitly specified rules as set by legislation.
- PA is a heaviest service industry, with a service production distributed in thousands of partially independent but at the same time collaborating agencies. This means that the Service Oriented Architecture (SOA) paradigm, which places the "service" notion at the core of development (e.g. Semantic Web Services) and is based on the idea of "loosely coupling" is particularly suitable and fits well with these institutional and structural characteristics of the PA domain.

The Governance Enterprise Architecture (GEA) effort [6] defines a conceptual generic domain model for PA. This model defines common aspects and generic features of the domain, with emphasis on service and process models. Moreover, it constitutes the basis for a top-level reference ontology of the eGovernment domain [7]. Such an ontology is generic enough to cover the overall eGovernment domain while at the same time specific enough to sufficiently model PA specific semantics.

In this paper, we extend our earlie[8-11] and present the GEA PA Service model in OWL. We present the process of "ontologizing" a conceptual model to create a formal domain description using a common ontology language, namely OWL-DL and discuss problems we identified in the process. Furthermore, a use case and a software application are presented to present the applicability, and practical value of the proposed ontology. The platform receives as input a citizen's profile and provides as output a set of public administration services that match with this profile. This functionality addresses the mapping between citizens' need/goal and available public services, which is a very demanding requirement in contemporary, complex, distributed and multi-layered administrative systems.

The rest of the paper is organized as follows: Section 2 describes related work in a new however very active field of using semantic models and technologies for eGovernment applications. In section 3 we briefly present the conceptual service model. Section 4 discusses how we developed the OWL version of this model, our modeling decisions and some lessons learnt. Section 5 describes the application architecture that uses our ontology. Finally, the conclusion and future work are given in section 6.

2 Related work

Several attempts were made to apply semantic web technologies in the eGovernment domain. The main objective in all cases is to improve the provision of public administration services by applying the semantic web technologies [12-15]. Some cases refer to attempts to build generic eGovernment representations and models that cover wide application areas and are not restricted to specific cases.

Ontologies play an important role in information systems engineering using semantic web technologies [16]. The DIP eGovernment ontology [17], which is stated to be "an extensive ontology that models a wide range of eGovernment and community services and information".

The ONTOGOV service ontology [18] is an eGovernment domain-specific service ontology, or a meta-ontology, proposed by the ONTOGOV IST project. The proposed ontology is heavily based on the two major generic service ontologies, namely OWL-S [19] and Web Service Modelling Ontology (WSMO) [20]. WebDG Ontologies [21] have been developed done under the Web Digital Government (WebDG) project. In the project ontologies were used to organize government information in order to make automatic composition feasible. WebDG attempts to present domain independent organization of service concepts in order to set up a Web Service-based environment for electronic service provision with special interest in automatic Web Service composition. Thus, WebDG aims to provide an infrastructure that is reusable in several different domains (e.g. e-banking, e-commerce) and avoids domain specific specialization or modelling.

The SAKE project deploys a holistic framework and supporting tools for an agile knowledge-based e-government using semantic technologies [22]. The eGov project [23-25] proposed a platform that included: governmental portals; the service repository and service creation environment; the Governmental Markup Language (GovML [26]); and the supporting network architecture.

The use of ontologies in reorganizing e-document management in public administration is presented in [27]. The usage of ontologies and semantic technologies for describing eGovernment services and can improve the management of changes [28]. Automated classification of citizens using an ontology is given in [29]. An intelligent search engine for an eGovernment application based on modeling the systems' electronic catalogue using an ontology is presented in [30]. An infrastructure for personalization of eGoverment services using ontology-based profiling of users citizens is given in [31]. An activity-based approach for the development and use of ontologies for eGovernment services is introduced in [32]. An eGovernment Business ontology was designed in [33] for sharing eGovernment business knowledge. A quality ontology for the adaptive evaluation of eGovernment services is given in [34]. Other domain specific ontologies include an ontology for modeling of life events [35], an ontology for an e-participation recourse center [36] and in an eGovernment virtual organization [37]. An application that creates eGovernment forms from semantic descriptions is given in [38]. A domain specific conceptual model for eGovernment services in the German federal state of Schleswig-Holstein and its ontology representation using WSML is given in [39]. An ontology-based model for multilingual knowledge management in information systems of European Union (EU) is presented in [40]. Moulin et al. [41] proposes the automated classification of citizens in knowledge bases. This classification can be applied to the main criteria of social care applications. A novel method for ontology development that combines ontology learning and social-tagging process is proposed in [42]. A prototype GIS application that combines spatial planning information in GML with XML and OWL is developed in [43]. The use of ontologies for electronically supporting and structuring public policies for interaction and collaboration among many heterogeneous government organizations (G2G collaboration) is presented in [44]. The authors in [45] present a process for building a domain ontology from scratch in public administration and give an example case by creating an Government Budgetary Ontology.

The above presented ontologies are modeling attempts of the basic entities of a specific PA domain (e.g. Justice) following an application-independent approach.

The Federal Enterprise Architecture (FEA) ontology [46] has been proposed by the USA CIO Council and consists of various approaches, models and definitions for communicating the overall organization and relationships of architecture components required for developing and maintaining a Federal Enterprise Architecture. The Federal Enterprise Architecture is being constructed through a series of "reference models" designed to facilitate cross-agency analysis and improvement. It is worth mentioning that with the exception of Business Reference Model (BRM), all the other FEA Reference Models cannot really be considered as PA domain specific as they are domain independent.

More details and analysis about eGovernment data models can be found in [47].

3 The Governance Enterprise Architecture (GEA) Service Object Model

In this section we briefly present the GEA service object model. GEA aims at introducing a consistent set of models that constitute the basis for top-level reference eGovernment domain ontology. A key aspect of GEA is that it attempts to be technology-neutral. This means that the GEA models may be applicable to different technological environments. A GEA overview can be found in [48]. The models are presented in detail in [6].

For the purpose of this paper, we focus on the GEA detailed service object model for service provision referred to in this paper as the PA Service Model for the sake of brevity. The overall object model is presented in Figure 1. A brief textual description follows.

Societal Entities (e.g. citizen, business) have *Needs* related to specific *Goals*. A *Societal Entity* requests a *Public Administration (PA) Service* to serve its *Goals*.

There are several types of *Social Entities* (e.g. legal entity, physical person). There are two categories of *Governance Entities* participating in service provision: *Political Entities* and *Public Administration Entities*. Based on the role, which *PA Entities* can acquire during the service execution phase, we identify four roles:

- *Service Provider* is the PA Entity that produces and provides the service to the Societal Entities (clients).
- *Evidence Provider* is the PA Entity that provides necessary Evidence to the Service Provider in order to execute the PA Service.
- *Consequence Receiver* is a third PA Entity that should be informed about a PA Service execution.
- *Service Collaborator* is the PA Entity that participates in the service execution and contributes to some part(s) of the service workflow.

Political Entities define *PA Services*. *PA Entities* through their role of *Service Provider* offer these services. *PA Services* are governed by *Preconditions* usually specified in Legal Acts - Laws. *Preconditions* set the general framework in which the service should be performed and the underlying business rules that should be fulfilled for the successful execution of the PA Service. *Preconditions* can be formally expressed as a set of clauses.

Public services need *Input* in order to be executed. *Input* is the complete set of information required to be checked or used in any way in order for a service to be executed. *Evidence placeholder* is the part of *Input* that contains *Piece of Evidence* that is the piece of information that the *Service Provider* should have access to in order to check the validity of *Preconditions*. An *Evidence Placeholder* usually "stores" several *Pieces of Evidence* and a specific *Piece of Evidence* that may be found in numerous different placeholders. For example, a citizen's age, serving as a *Piece of Evidence* for a service that sets age limitations in its *Preconditions*, can be contained in the ID card, the passport or the birth certificate. These are considered as alternative

Evidence Placeholders. There are many cases where the *Evidence Placeholders* are provided by *PA Entities* (Evidence Providers).

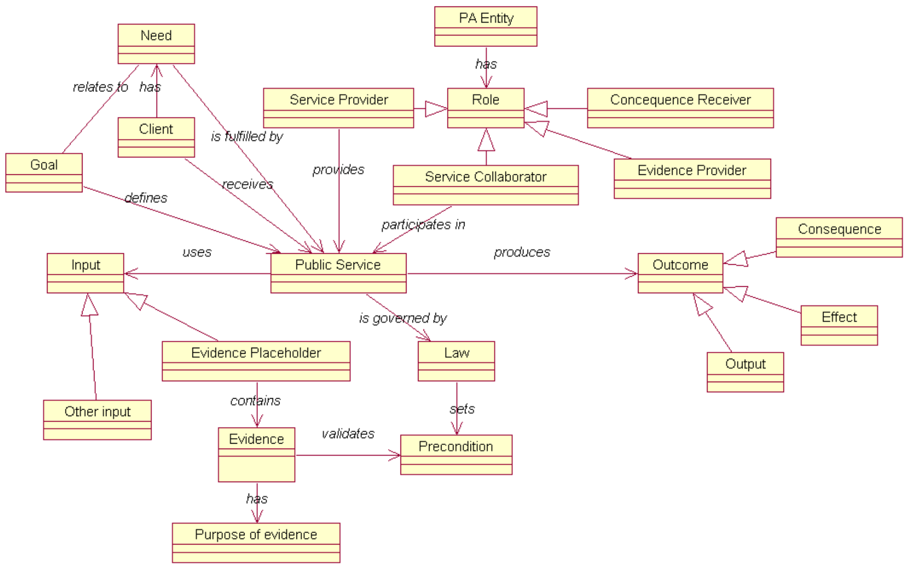


Figure 1 The GEA PA Service Object Model.

Various types of *Evidence Placeholders* exist. For example, they may be physical (e.g. documents) and electronic (e.g. databases, XML documents). Information that is used by the service for other purposes than preconditions' validation is not considered evidence and is modeled here as *Other Input* (e.g. the applicant's address to be used for communicating a document/decision after the service execution).

In each service, a *Piece of Evidence* has a *Purpose*. The *Purpose* is the underlying business logic that explains the reason for which the service provider wants to have access to the specific piece of information. For example, the purpose of the ID card number (evidence), which is usually asked to service applicants, serves for checking and validating the identity of the applicant.

The *Outcome* refers to the different types of results that a PA Service may have. GEA defines three types of *Outcome*:

- *Output*, which is the documented decision of the *Service Provider* regarding the service asked by a *Societal Entity*. This reaches the client in the form of an administrative document/decision.

- *Effect*, which is the change in the state of the real world (e.g. transfer money to an account) caused by the execution of a service. In the PA domain, the service *Effect* is the actual permission, certificate, restriction or punishment the citizen is finally entitled to. In cases where administration refuses the provision of a service, there is no *Effect*.

- *Consequence*, which is information about the executed PA Service that needs to be forwarded to interested parties. As an example, in Greece someone can adopt a child through a service provided by the Prefecture of the foster parents' residence. The municipalities where the foster parents were born will then have to be informed about the event, in order to update their population registries. This is the *Consequence* of the adoption service.

The above objects are compatible with the concepts defined by the OASIS reference model for SOA [49]. The OASIS model defines the concept of Real-World Effect. One or more Real-World Effects are produced after service invocation. These may include:

- Information returned in response to a request for that information (superclass of the GEA *Output* object).
- A change to the shared state of defined entities (that maps to the GEA *Effect* object*)*
- Or a combination of the above (that is similar to the GEA *Consequence* object).

Additional objects and relations of the PA service are depicted in Figure 2. Their description follows. PA Services are categorized in several *PADomains* (e.g. Health, Transportation). Each *Domain* object is divided into several *SubDomain* objects (e.g. Domain Transportation has SubDomains Ground Transportation, Air Transportation and Water Transportation). The categorization that we have used is based on the categorization proposed in the Federal Enterprise Architecture (FEA) [46, 50]. The PA Entities belong to an *Administrative Level* (e.g. municipality, regional). Each PA service is offered at a distinctive *Administrative Level*. Each PA Service is also offered at a specific *Location* (electronic or physical or both). PA services have generic service types. In GEA the following five generic types of public services are identified:

- *Declaration*: Through providing public services of declaration type, public administration declares and registers changes in the world state (e.g. marriage)
- *Certification*: Through certifications public administration certifies existing states of the world (e.g. issuing birth certificate).
- *Control*: PA uses a specific type of public service in order to address this: “Control”. As the offender tends to hide his/her behavior from PA, the most ordinary type of control is inspections on a periodic or on an impromptu basis.
- *Authorization*: Through this type of public services public administration realizes both permissive and support goals.
- *Production*: Public administration uses this type of public services in order to produce new public services.

The execution of a PA service is expected to have an *Effect type*. At the top level, there are three types of Effects have been identified:

- Safeguard the Social Contract; meaning maintain the peaceful coexistence amongst the members of society.
- Promote Sustainable Development; meaning providing for macro-economic development taking into account sustainability concepts (e.g. environment).
- Provide for Social Welfare; meaning enhancing social cohesion by coping with exclusion and poverty.

There are cases where public service execution depends on the citizen’s (client) profile. For example a benefit issuance PA service for single mothers. In order to model such cases the above-described service model can be extended to take into account the dependence of a citizen's profile with PA service execution. This is done with the use of a *Profile Descriptor* object. This object represents the concept of a citizen's profile characteristics that can be used to validate the execution *Preconditions* of a certain *PA Service*. Therefore decide if she/he is eligible for this service. The *Profile Descriptor* object has several types like *Age Category ,Health status* and many other profile characteristics. For example a Parking License Issuance for disabled people PA service has the precondition that the client has to have a *Profile Descriptor* of *Health Status* value disabled. The *Precondition* object defines the profile characteristics for a given PA service.

Recently the Open Group has presented a generic SOA ontology in OWL [51]. This ontology was developed in order to be a basis for model-driven implementation, thus it should be applied to particular usage domains. The GEA service object model can be seen as a "particularization" of this generic SOA model. GEA defines a particular model applicable only to PA services and it is strongly oriented to this domain. The mappings between generic SOA objects and GEA objects are straightforward as it can be seen in Table 1. This proves that our conceptual modeling of the Public Service is compatible and is in accordance with the generic service model developed by the Open Group.

Table 1. GEA-SOA Open Group Concepts

GEA object	**SOA object**	**Comments**
Public Service	Service	Generic Service object
Service Provider	Actor	Linked to the Service with provides property. Subclasses of the Actor class are Human Actor, Organization Actor, and Technology Actor.
Client	Actor	Linked to the Service with consumes property.
Outcome	Effect	A service has effects. These comprise the outcome of the service, and are how it delivers value to its consumers.
Input, Evidence Placeholder, Output	Information Item	An information item is a thing that is known about some other thing. The SOA ontology includes the Information Item class, whose instances are such pieces of information.
Evidence	Information Type	An information item may have one or more information types
Effect	Change	A change to some thing conveys the idea that the thing is different beforehand and afterwards. For example in a car-wash service the change after service execution is that the car is clean.
Consequence	Event	An event is something that happens, to which an activity may respond. An event can be an effect of an action. In the car-wash service example the money given by the customer of the service is an Event.
Law	Rule	A rule applies to a service.

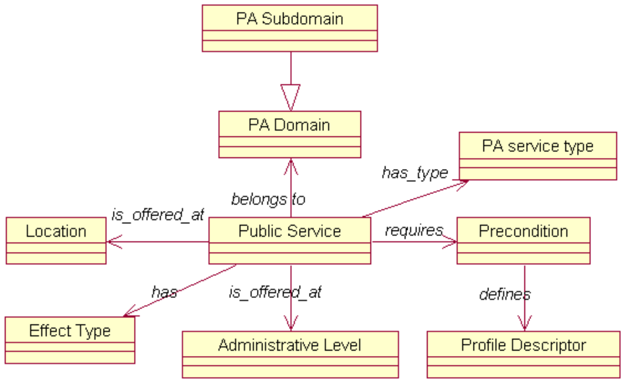


Figure 2 GEA PA Service extended model objects

4 PA Domain Reference Ontology in OWL

The GEA service object model in Figure 1 was shown in a UML class diagram. It is obvious that such a model can be implemented using a relational database. Such an approach would be complex since it would not exploit the advantages of declarative knowledge representation. The main requirement today is to be able to share information through the web for both humans and machines. We have decided to express the GEA service object model in an ontology language.

OWL DL [52] was the obvious choice for several reasons; since 2004, OWL DL is an active recommendation of W3C group and various examples of models expressed in OWL exist. Several one-to-many relations exist in the GEA Service model. Therefore, the full expressiveness of cardinality restrictions requires the use of OWL DL instead of OWL Lite. Another important point that was taken into account is the existence of OWL DL reasoners. OWL Full reasoners do not yet exist. The GEA ontology has been created using the Protégé tool with the OWL plug-in [53]. A part of the GEA model class hierarchy in OWL DL is shown in Figure 3. The basic modeling principles followed were:

- The GEA service object model entities were expressed in *owl: Class elements*. We have followed the naming convention that every ontology Class is named using the prefix GEA e.g. Public Service object is the GEA_Public_Service class.The classes in this model refer to different objects and should not be allowed to overlap. Therefore all the classes in the same hierarchy level were declared *owl: disjointWith*.

- The relations between entities were expressed in *owl: ObjectProperty* metaclasses. In cases were the relations were one-to-one, they were expressed in *owl: FuctionalProperty* metaclasses.

- OWL does not make the Unique Name Assumption (UNA). Just because two names are different, it does not mean that they refer to different individuals. Therefore for individuals that belonged to the same class the elements *owl:distinctMembers* were used in combination with *owl:AllDifferent* to define a set of mutually distinct individuals.

- In the GEA service object model several has-type relations exist between entities. In some cases these were modeled in OWL as *rdfs: subClassOf*. In some other cases these relations were modeled using object properties. For example, the Output object was not modeled as a subclass of the Outcome object. This is due to the fact that the Output object in Public Administration is actually an Evidence Placeholder. Two new object properties were created; *hasOutput* with domain *GEA_Public_Service* and range *GEA_Evidence_Placeholder*, and its inverse property *isOutputOf*. The *Effect* and *Consequence* objects were modeled as subclasses of the *Outcome* class.

- The PA entity object in the GEA model has three distinct roles, *ServiceProvider*, *EvidenceProvider* and *ConsequenceReceiver*. These roles are depicted in OWL using three object properties. For example the *ServiceProvider* role is modeled using the *owl: ObjectProperty providesServices* with domain *GEA_PA_Entity* and range the *GEA_Public_Service* class.

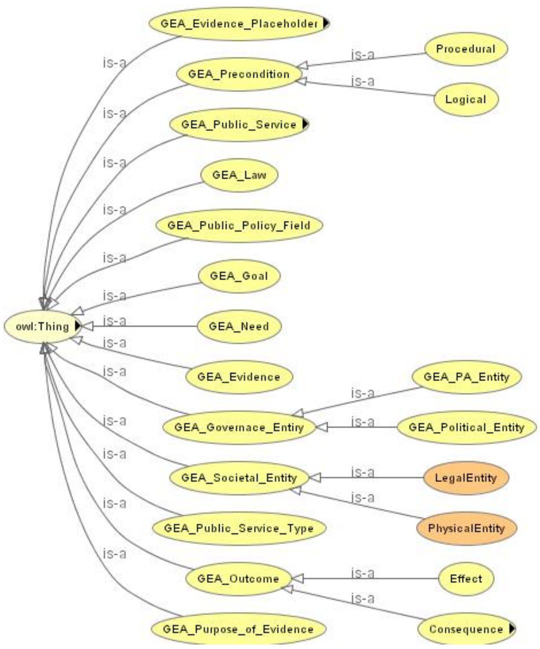


Figure 3 The GEA ontology.

Table 2. GEA_Public_Service Class in OWL Abstract syntax.

Class(GEA_Public_Service partial owl:Thing restriction(hasPublicServiceType allValuesFrom(oneOf(GEA_Authorization GEA_Control GEA_Production GEA_Certification))) restriction(hasAdministrationLevel cardinality(1)) restriction(hasEffectType cardinality(1)) restriction(hasLocation minCardinality(1)) restriction(hasPADomain cardinality(1)) restriction(hasClientType minCardinality(1)) restriction(hasPASubDomain cardinality(1)))

The classes that were not shown in Figure 3 are given in Figure 4. These classes represent the property descriptors of the *GEA_Public_Service* class. Due to visualization reasons only the classes that are used in the application appear. The object properties that connect the *GEA_Public_Service* class with other classes of the ontology are also shown in Figure 4. The *GEA_Public_Service* Class definition with the object property restrictions is shown in Table 2 in OWL Abstract Syntax [54]. Below follows a brief description of these classes.

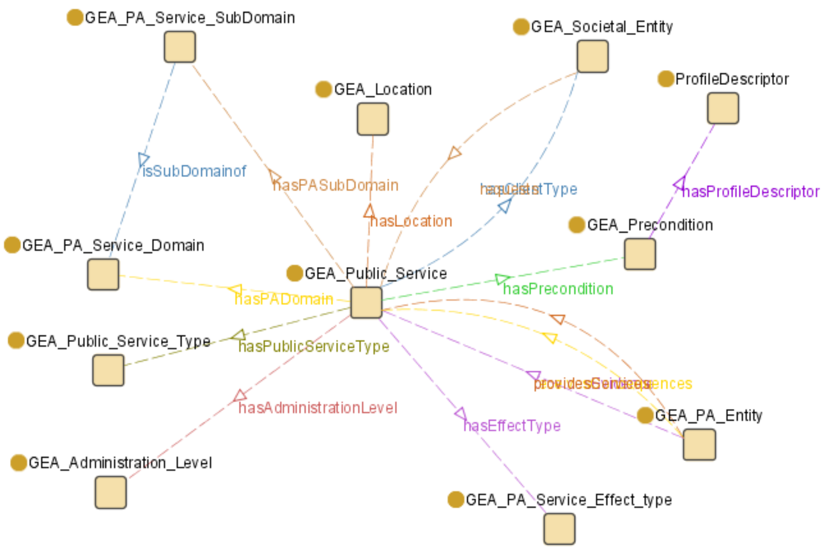


Figure 4 The GEA_Public_Service Class

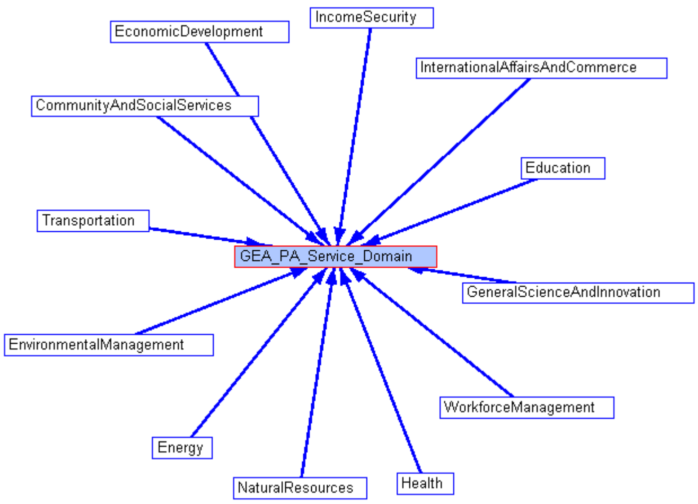


Figure 5 The GEA_PA_Service_Domain individuals

A common problem in OWL is that we need to refer to some objects sometimes as individuals and other times as classes. OWL DL does not support the usage of the same object as both class and instance. A way to overcome this problem is to use a certain design pattern called "Class-

Instance Mirror" [46]. This design pattern is recommended by the W3C Semantic Web Best Practices Group[55]. This design pattern has been used for the PA_Service_SubDomain object described below. *GEA_PA_Service_Domain* class represents the different PA domains. It has been populated with the following individuals; *CommunityAndSocialServices*, *EconomicDevelopment, Education, Health, Transportation, GeneralScienceAndInnovation, IncomeSecurity, InternationalAffairsAndCommerce EnvirionmentalManagement*, *Energy*, *NaturalResources* and *WorkforceManagement*. These individuals are declared different using the *owl:allDifferent* element. One may notice that these domains correspond to USA Federal Enterprise Architecture (FEA) [46, 50] Business Reference Model *ServicesforCitizens* object. The *GEA_PA_Service_Domain* class individuals are presented in Figure 5.

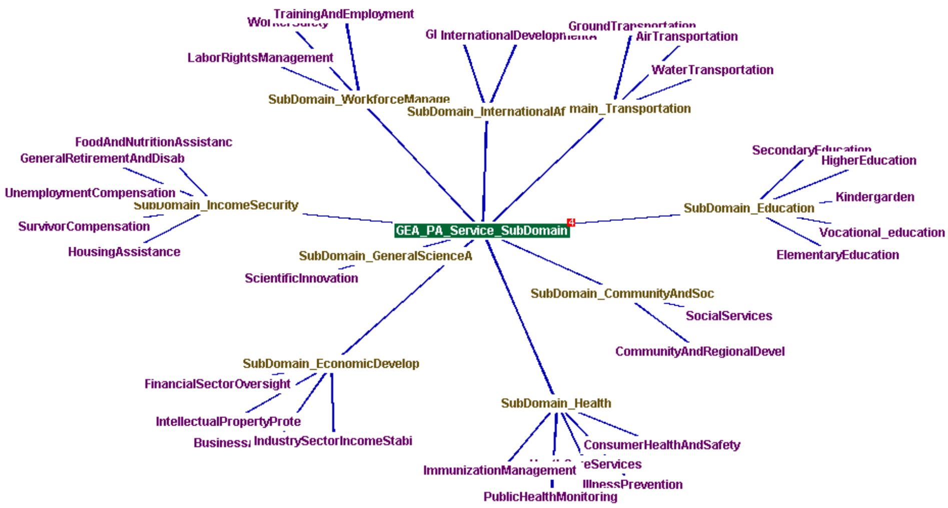
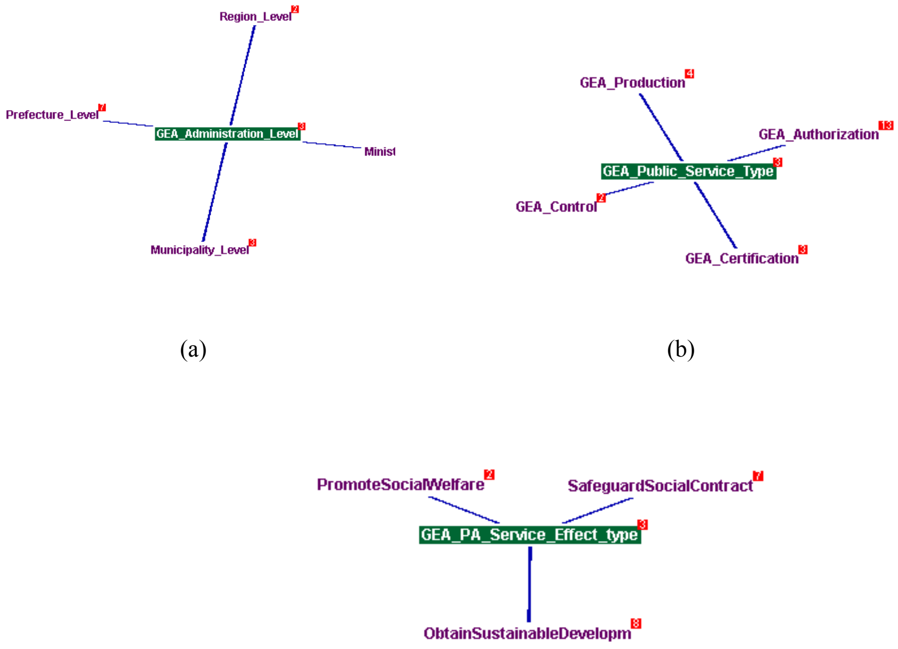


Figure 6 The GEA_PA_Service_SubDomain subclasses and individuals.

Each of these main domains has a number of subdomains. These are represented by the *GEA_PA_Service_SubDomain* Class (equivalent to *SubFunction* class in FEA ontology). Subclasses of this class have been created and all are modeled as an *owl: EquivalentClass*. Each subclass represents a PA domain and the individuals who belong to the class represent the PA SubDomains (e.g. Domain *Transportation* has SubDomains *Ground Transportation, Air Transportation* and *Water Transportation*). The *GEA_PA_Service_SubDomain* Class with subclasses and individuals is shown in Figure 6. The *GEA_Administration_Level* class is populated by four individuals *Ministry_Level*, *Prefecture_Level*, *Municipality_Level* and *Region_Level*. These individuals represent the different administration levels in Greece. In other countries other individuals may be used (e.g *Federal_Level*). This class represents the unique administration level at which each public service is offered. The *GEA_Public_Service_Effect_Type* class represents at an abstract and high-level, the three distinct effect types achieved by a public service. The individuals that belong to that class are *ObtainSustainableDevelopment*, *PromoteSocialWelfare* and *SafeguardSocialContract* [6]. The *GEA_Public_Service_Type* class represents the concept of the generic PA service type. This class is populated with five individuals: *GEA_Production*, *GEA_Control*, *GEA_Certification* , *GEA_Authorization* and *GEA_Declaration*. Figure 7 depicts the above-described classes and individuals.



(a) (b)

(c)

Figure 7. The individuals of the (a) GEA_Administration_Level class (b) the GEA_Public_ Service_Type class and (c) the GEA_PA_ Service_Effect_Type

The *GEA_Location* represents the physical or electronic location where the public service is offered. For the physical location, a top-level, location ontology can be imported.

All the above public service descriptors are linked with individuals from the *GEA_Public_Service* class using *owl:ObjectProperty* elements with domain *GEA_Public_Service* and range the corresponding public service descriptor class.

ProfileDescriptor class (Figure 8) represents the concept of a citizen's profile characteristics that can be used to validate if she/he is eligible for a certain public service. One may notice that this class contains only the information needed for service retrieval while other information like user name is not required. This is due to the fact that public services set preconditions for eligible citizens based on user profile. For example in order for a citizen to be eligible for a driving license service he/she has to be an adult. We have modeled this fact by creating the *ProfileDescriptor* class. In order to model the different profile characteristics subclasses of this class were created. These subclasses are populated with individuals that represent the different categories of this subclass. For example the individuals Male and Female belong to the Gender subclass. Some subclasses of this class are:

- *AgeCategory* (adult, adolescent, senior)
- *Healthstatus* (normal, disabled e.t.c)
- *FamilyStatus* (divorced, married, single, widow)

- *EducationLevel* (technical, higher, elementary)
- *Gender* (Male, Female).
- *Citizenship* (EU, Greek, Other).

One may notice that some of the above subclasses could be also modeled as enumerated datatypes. We have chosen to use the above-described approach due to the fact that there are not available OWL DL reasoners that support datatype reasoning. OWL 2 [56], the next version of OWL, supports datatype reasoning

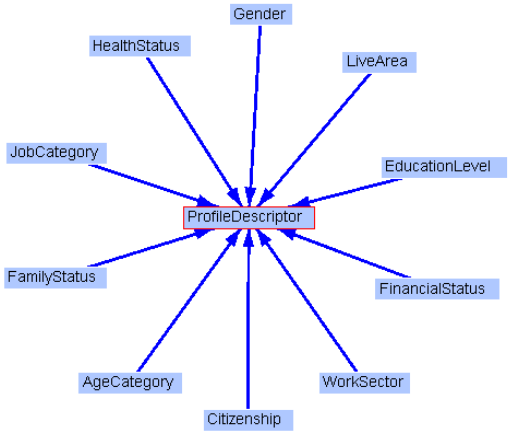


Figure 8 The Profile Descriptor Class with subclasses

The above characteristics were modelled in order to serve as an example for our application. The *ProfileDescriptor* class can be easily expanded with new subclasses that will model other profile characteristics. These depended on the requirements that a PA service may impose. For example a PA service that provides provisional driving licenses in Ireland would require the modelling of a *Live Area* profile characteristic.

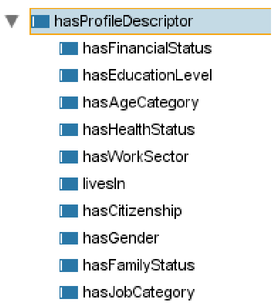


Figure 9 The Profile Descriptor Class properties

Preconditions are modeled in the ontology using the *GEA_Precondition* class. To model a precondition in OWL restrictions on object properties can be used. This can be accomplished by using necessary and sufficient conditions in class description. Such a class is called a *defined class* in OWL terminology. This means that if an individual fulfils this condition then it must be a member of this class.

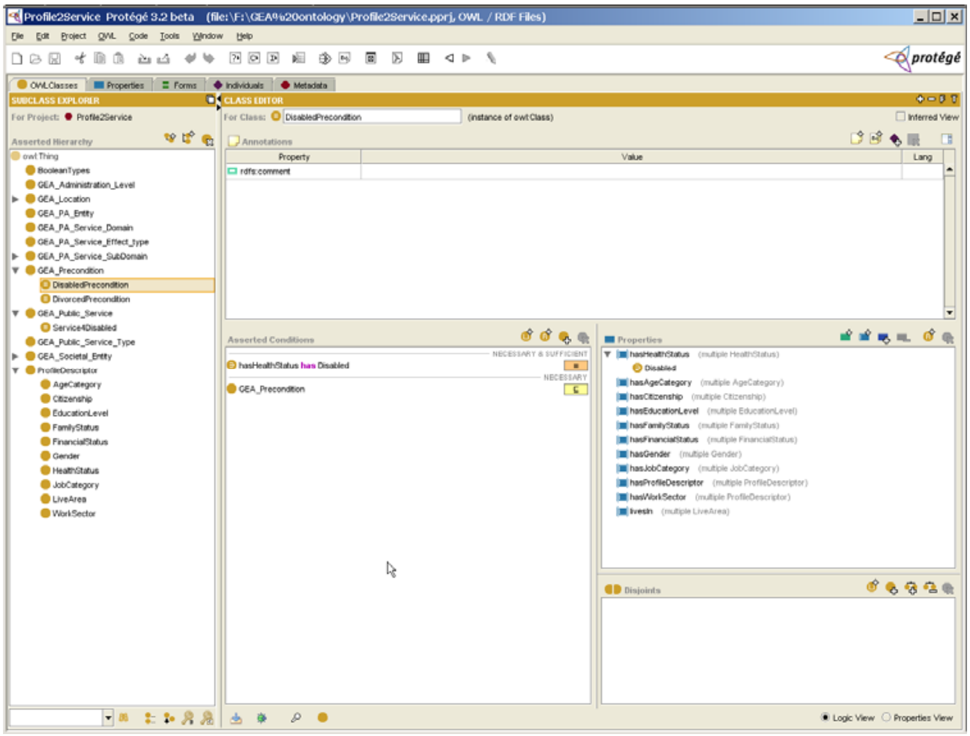


Figure 10 DisabledPrecondition Class.

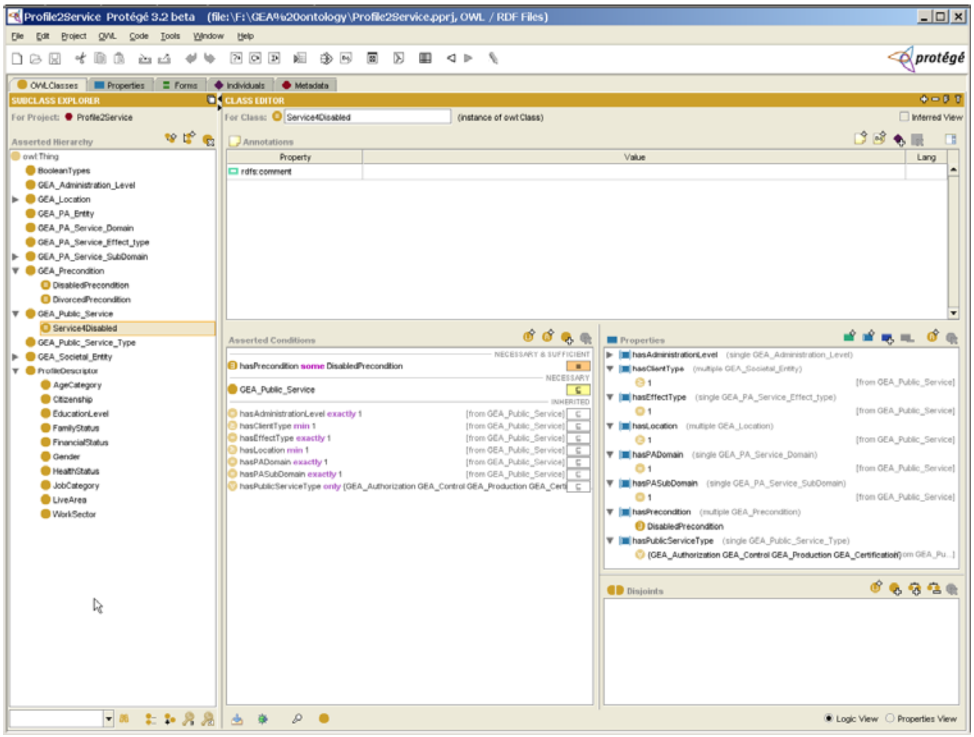


Figure 11 Service4Disabled Class.

GEA_Precondition class is linked with the *ProfileDescriptor* class with the object property *hasProfileDescriptor* which has domain *GEA_Precondition* and range *ProfileDescriptor*. The subproperties of *hasProfileDescriptor* link individuals from *GEA_Precondition* class to corresponding subclasses of *ProfileDescriptor*. For example object property *hasHealthStatus* is declared to have domain the *GEA_Precondition* class and range the *HealthStatus* class. These properties can only have a unique value, therefore they are declared functional (Figure 9).

Subclasses of the *GEA_Precondition* class are created for every profile case. For example *DisabledPrecondition* class is declared to be a defined class of the *hasValue* restriction that all individuals that belong to this class have the value *disabled* in *hasHealthstatus* property. *DisabledPrecondition* class is shown in Figure 10. *GEA_Public_Service* class is connected with *GEA_Precondition* with the object property *hasPrecondition*, which has domain *GEA_Public_Service* and range *GEA_Precondition*. Every *GEA_Public_Service* subclass is created using restrictions on this property. For example a public service that has a benefit for disabled persons is declared having value restrictions in *hasPrecondition* of *DisabledPrecondition*. Figure 11 shows the *Service4Disabled* Class.

Concluding the ontology development process, the lessons we have learned can be given as guidelines:

- Consulting a domain expert can speed up the whole process and ensure that domain knowledge representation is accurate.
- Technology restrictions that apply make the selection of the modeling language an important step. This depends not only on expressive power but also on the availability of the supporting tools like ontology editors and reasoners.
- The representation of UML objects into OWL classes is not always straightforward. In some cases object properties can be used instead.
- The lack of datatype reasoning in OWL DL increases the complexity of the modeling process.
- Valuable input can be provided from similar design efforts. In our case the SOA [51] and the FEA [46] ontologies provided useful modeling patterns.
- Ontology validation is the final step of the development process. This consists of two parts: the consistency checking that is actually performed by Protégé during design time and the validation that is made using the application in run time.

5 Use case: matching citizens' profiles to public services

The ontology presented earlier may serve as a generic PA domain ontology in a variety of applications. In a nutshell, in our research agenda, we perceive the GEA PA domain ontology as the basic knowledge infrastructure for eGo. A sample application for finding public services which match to specific citizens' profile is given in this section.

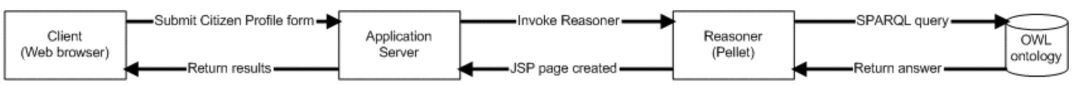


Figure 12. Sample application architecture.

The system architecture is shown in Figure 12. It consists of an application server, a reasoner and an OWL file which plays the role of the knowledge base. There all services are represented using the ontology discussed in the previous part. The users access the application through a common Internet browser.

The system architecture employed is server-side; therefore the client shows only the form and the results page. The application server uses the data given to invoke the reasoner. The reasoner sends various SPARQL [57] queries to the knowledge base. Using the answers returned the web server creates the results page. The extracted results contain a list of the public services that match the selected profile. Specifically, the application server used was Apache Tomcat. The reasoner selected is Pellet [58]. Pellet is an open source OWL DL reasoner that can be used in conjunction with Jena. Pellet provides support for SPARQL. At the front-end, when the user enters the first page of the application, a form that contains fields about his/her profile appears. The user fills in the form the characteristics of his/her profile based on profile descriptor class.

There are two types of users in the system:

- Citizens who search for services that match their profile and domain experts.
- Domain experts who are civil servants with the task to create and update the content of the knowledge base.

Services are added and maintained using the Protégé open source tool [53]. In particular, domain experts use the Protégé Web Browser, which is a Java-based web application that allows creating, updating and sharing of OWL knowledge bases.

In our use case, we assume that a citizen who is disabled wants to find all the public services that are suited for him/her. Let's assume that the municipal authority has – among others – a public service for the issuance of a free parking license for a disabled person (*DisableParkingLicenceIssuance*) and another service that issues a monthly benefit to disabled persons (*DisabledBenefitIssuance*). These individuals as modeled using our ontology in Protégé are depicted in Figure 13.

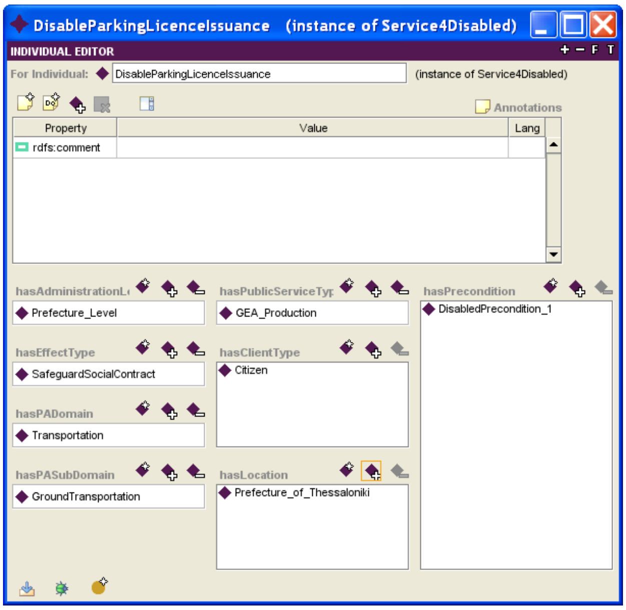
The semantic discovery of the public services that match the specific profile is executed as a two-step process. The first step is to find all the preconditions that match the user's profile. In the disabled user case these are the individuals of the *DisabledPrecondition* class. We assume that such an individual is DisabledPrecondition1. Therefore a first SPARQL query is send to the reasoner, which is given in Table 3 below:

Table 3. SPARQL query for finding the Preconditions that match the specified profile.

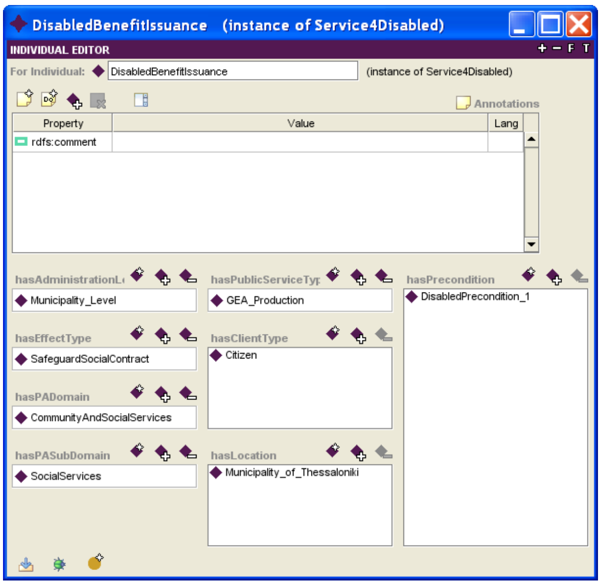
```
PREFIX gea:  < http://localhost/Profile2service.owl #>
PREFIX rdf:  http://www.w3.org/1999/02/22-rdf-syntax-ns#
SELECT  ?Precondition
WHERE { ?Precondition rdf:type gea: Precondition.
? Precondition gea:hashealthstatus gea:Disabled. }
```

The reasoner answers that the Precondition is *gea: DisabledPrecondition1*. The first step ends when all the needed *GEA_Precondition* individuals for the given profile values have been identified. Then a second querying step is triggered.

During the second step, the reasoner creates a new query that searches for public services that satisfy all the *GEA_Precondition* individuals found through the execution of the first querying step. The goal is to find all the public service individuals that are linked to individuals of the DisabledPrecondition class using the *hasPrecondition* property. Again a new SPARQL query is sent to reasoner (Table 4). In this way, the reasoner returns the services that are found to match the selected profile.



(a)



(b)

Figure 13. The DisableParkingLicenceIssuance (a) and DisabledBenefitIssuance (b) Individuals in Protégé

Table 4. SPARQL query for identifying services with hasPrecondition DisabledPrecondition

```
PREFIX gea:  < http://localhost/ Profile2service.owl #> ·
PREFIX rdf:  http://www.w3.org/1999/02/22-rdf-syntax-ns#
SELECT  ?Service
WHERE { ?Service rdf:type gea:GEA_Public_Service.
?Service gea:hasPrecondition  gea: DisabledPrecondition1.}
```

In the example case the reasoner returns the individuals of the *Service4Disabled* class e.g *DisableParkingLicenceIssuance* and *DisabledBenefitIssuance*. If the user had specified more profile characteristics then similar queries would have been sent to the reasoner based on these profile characteristics. For example if the user was divorced then additionally all the public services that are addressed to divorced citizens would have been discovered.

6 Conclusions and Future Work

The GEA PA service model can be used as the basis for a top-level eGovernment domain ontology. In this paper we have expressed this generic PA service model in OWL DL. This PA domain reference ontology can play an important role in semantic web applications for eGovernment. This ontology may serve as the knowledge base in a variety of semantic web applications for Public Administration. For example an application that helps a citizen to locate all the PA services offered in specific locations or specific PA administrative units (e.g. municipality of Thessaloniki) can use the same ontology populated with PA service individuals. Our intention was to create a generic ontology that can be easily expanded to model most of the PA service provision aspects in any country of the world. Ontologies at a local or at a national level can be created by importing this generic ontology and by adapting it to match the current PA requirements. These local versions will be populated with the appropriate individuals for every case. In this way the semantic web vision of creating and sharing knowledge worldwide through the web is becoming reality.

From a technical perspective, OWL has the expressiveness to represent the Public Administration domain specific knowledge. Protégé is yet the most efficient freeware ontology editor, but it is not bug-free. The ontology validation with an external DIG [59] reasoner is a useful feature. The use of Pellet [58] for the required reasoning has offered several advantages. Pellet provides support for DIG compliant reasoning and for several query languages (SPARQL). Pellet supports also all the features proposed in OWL 1.1 [60]. SPARQL is a relatively new W3C Recommendation, which provides an easy to use SQL-like RDF query language. The lack of update commands is still a disadvantage. The technologies used offered the capabilities to create a semantic application but most of them are still on-going research projects.

The presented application applies semantic service discovery based on citizen profile. The citizen selects his/hers profile and the application uses reasoning to find the matched public services. The GEA PA service object model is proven to have the adequate expressiveness for such an application. In a second stage the application should be able not only to discover, but also to execute the public services found in this way. In order to achieve this, an approach based on a semantic web services framework and execution environment is required (e.g. WSMO, WSMX). This is part of our current work based on semantic web services technologies.

Acknowledgements

This paper is supported in part by the SemanticGov project (FP6-2004-IST-4-027517; http://www.semantic-gov.org) and by Science Foundation Ireland (SFI) under Grant No. SFI/02/CE1/I131.

References

1. Stojanovic, N., et al. Fostering self-adaptive e-government service improvement using semantic technologies. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford, CA).

2. Lytras, M.D., The semantic electronic government: Knowledge management for citizen relationship and new assessment scenarios. Electronic Government, 2006. 3(1): 5-17.

3. Fass, L.F. The semantic web, e-Government and the digital divide. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford, CA).

4. Traunmuller, R. and Wimmer, M., KM for Public Administration: Focusing on KMS Feature Requirements, in *PAKM 2002*, D. Karagiannis and U. Reimer, Editors. 2002, Springer-Verlag: Berlin Heidelberg 2002. p. 314-325.

5. Klischewski, R. and Lenk, K., Understanding and Modelling Flexibility in Administrative Processes, in *Electronic Government*. 2002. p. 129-136.

6. Peristeras, V., The Governance Enterprise Architecture - GEA - for Reengineering Public Administration, PhD Dissertation, in *Business School Department*. 2006, University of Macedonia: Thessaloniki.

7. Grunlund, A. What's In a Field – Exploring the eGoverment Domain. in 38th Hawaii International Conference on System Sciences. 2005.

8. Peristeras, V. and Tarabanis, K. Reengineering the public administration modus operandi through the use of reference domain models and Semantic Web Service technologies. in 2006 AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford University, California, USA): AAAI.

9. Peristeras, V. and Tarabanis, K., The Governance Enterprise Architecture (GEA) Object Model, in *5th IFIP International Working Conference, Knowledge Management in Electronic Government:KMGov 2004, Krems, Austria, May 26-28 2004. Proceedings*, Maria A. Wimmer, Editor. 2004, Springer-Verlag Heidelberg. p. 101-110.

10. Peristeras, V. and Tarabanis, K., Advancing the Government Enterprise Architecture (GEA): The Service Execution Object Model, in *Electronic Government: 3rd International Conference, EGOV 2004, Zaragoza, Spain, September 2004. Proceedings*, R. Traunmueller, Editor. 2004, Springer-Verlag Heidelberg. p. 476-482.

11. Goudos, S.K., Loutas, N., Peristeras, V., and Tarabanis, K. Public Administration Domain Ontology for a Semantic Web Services EGovernment Framework. in IEEE International Conference on Services Computing, (SCC 2007) 2007. (Salt Lake City, USA).

13. Sacco, G.M. User-centric access to e-government information: E-citizen discovery of e-services. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford, CA).

14. Ilgar, E., Oldenhuizing, J., and Mika, P. Let the Citizen Speak: A demand-driven e-government portal using Semantic Web Technology. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford, CA).

15. Vitvar, T., et al. Infrastructure for the Semantic Pan-European E-government Services. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford University, California, USA).

16. Tran, T., Lewen, H., and Haase, P. On the role and application of ontologies in information systems. in Proceedings of IEEE International Conference on Research, Innovation and Vision for the Future, RIVF 2007. 2007. (Hanoi).

17. DIP Project, D9.3 e-Government ontology. 2004.

18. Tambouris, E., et al., Ontology-enabled E-gov service configuration: An overview of the OntoGov project. Lecture Notes in Artificial Intelligence 2004. 3035: 106-111.

19. OWL-S Services Coalition. OWL-S: Semantic Markup for Web Services W3C Member Submission 2004 [cited March 2005]; Available from: http://www.w3.org/Submission/OWL-S

20. Roman, D., et al., Web Service Modeling Ontology. Applied Ontology, 2005. 1 77-106

21. Medjahed, B., Rezgui, A., Bouguettaya, A., and Ouzzani, M., Infrastructure for E-Government Web Services. IEEE Internet Computing Magazine, 2003. 7(1): 58-65.

22. Stojanovic, N., Mentzas, G., and Apostolou, D. Semantic-enabled agile knowledge-based e-government. in AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG). 2006. (Stanford, CA).

23. eGOV. Project IST-2000-28471[online]. 2003 [cited; Available from: http://www.egov-project.org.

24. Tambouris, E. An Integrated platform for Realising Online One-Stop Government: The eGov Project. in Proceedings of DEXA International Workshop "On the Way to Electronic Government". 2001. (Los Alamitos, CA): IEEE Computer Society Press.

25. Tambouris, E., Manouselis, N., and Costopoulou, C., Metadata for digital collections of e-government resources. Electronic Library, 2007. 25(2): 176-192.

26. Kavadias, G. and Tambouris, E., GovML: A markup language for describing public services and life events. Lecture Notes in Artificial Intelligence 2003. 2645: 106-115.

27. Klischewski, R., Ontologies for e-document management in public administration. Business Process Management Journal, 2006. 12(1 SPEC. ISS.): 34-47.

28. Stojanovic, L., Abecker, A., Stojanovic, N., and Studer, R., On managing changes in the ontology-based e-government. Lecture Notes in Computer Science, 2004. 3291: 1080-1097.

29. Moulin, C., Bettahar, F., Barthès, J.-P., and Sbodio, M., Ontology Based Categorization in eGovernment Application, in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*. 2007. p. 1153-1160.

30. Markellos, K., Sakkopoulos, E., Sirmakessis, S., and Tsakalidis, A., Semantic web search for e-government: The case study of intrastat. Journal of Internet Technology, 2007. 8(4): 457-468.

31. Grandi, F., et al., Semantic Web techniques for personalization of eGoverment services. Lecture Notes in Computer Science, 2006. 4231 LNCS: 435-444.

32. Klischewski, R. and Ukena, S. An Activity-Based Approach towards Development and Use of E-Government Service Ontologies. in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS41). 2008.

33. Xiao, Y., Xiao, M., and Zhao, H. An ontology for e-government knowledge modeling and interoperability. in Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM 2007). 2007. (Shanghai).

34. Magoutas, B., Halaris, C., and Mentzas, G., An ontology for the multi-perspective evaluation of quality in e-Government services. Lecture Notes in Computer Science, 2007. 4656 LNCS: 318-329.

35. Trochidis, I., Tambouris, E., and Tarabanis, K. An ontology for modeling life-events. in 2007 IEEE International Conference on Services Computing, SCC 2007. 2007. (Salt Lake City, UT).

36. Wimmer, M.A. Ontology for an e-participation virtual resource centre. in Proceedings of the 1st international conference on Theory and practice of electronic governance. 2007. (Macao, China): ACM.

37. Tang, X. and Yang, F. A study on dynamic ontology for information integration in E-governmental virtual organization. in Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM 2007). 2007. (Shanghai).

38. Stadlhofer, B. and Salhofer, P. Automatic generation of e-government forms from semantic descriptions. in Proceedings of the 1st international conference on Theory and practice of electronic governance. 2007. (Macao, China): ACM.

39. Bednar, P., et al., Semantic integration of eGovernment services in Schleswig-Holstein. Lecture Notes in Computer Science, 2008. 5184 LNCS: 315-327.

40. Segev, A. and Gal, A., Enhancing portability with multilingual ontology-based knowledge management. Decision Support Systems, 2008. 45(3): 567-584.

41. Moulin, C., Bettahar, F., Barthes, J.P., and Sbodio, M.L., Ontology based categorization in eGovernment application. Lecture Notes in Computer Science, 2007. 4803 LNCS(PART 1): 1153-1160.

42. Stojanovic, L., Stojanovic, N., and Ma, J., An approach for combining ontology learning and semantic tagging in the ontology development process: EGovernment use case. Lecture Notes in Computer Science, 2007. 4831 LNCS: 249-260.

43. Boer, A., Van Engers, T., Peters, R., and Winkels, R., Separating law from geography in GIS-based eGovernment services. Artificial Intelligence and Law, 2007. 15(1): 49-76.

44. Loukis, E.N., An ontology for G2G collaboration in public policy making, implementation and evaluation. Artificial Intelligence and Law, 2007. 15(1): 19-48.

45. Brusa, G., Laura Caliusco, M., and Chiotti, O., Towards ontological engineering: A process for building a domain ontology from scratch in public administration. Expert Systems, 2008. 25(5): 484-503.

46. TopQuadrant, FEA Refererence Model Ontologies (FEA RMO) v1.1. 2005.

47. Peristeras, V., Tarabanis, K., and Goudos, S.K., Model-driven eGovernment interoperability: A review of the state of the art. Computer Standards & Interfaces. In Press, Accepted Manuscript.

48. Peristeras, V. and Tarabanis, K. Governance Enterprise Architecture (GEA): Domain Models for e-Governance. in 6th ACM International Conference on Electronic Commerce (ICEC'04). 2004. (Delft, The Netherlands).

49. OASIS. Reference Model for Service Oriented Architecture 1.0. OASIS Standard, 2006 12 October 2006 [cited; Available from: http://docs.oasis-open.org/soa-rm/v1.0/.

50. CIO Council. Federal Architecture Enterprise Framework v.1.1. 1999 Sep. 99 [cited 5 Sep. 2002]; Available from: http://www.cio.gov/Documents/fedarch1%2Epdf.

51. The Open Group. SOA Ontology Draft 2.0 Draft Technical Standard 2008 14 Jul. [cited; Available from: http://www.opengroup.org/projects/soa-ontology/doc.tpl?CALLER=index.tpl&gdid=16940.

52. Dean, M., et al. . OWL web ontology language reference. W3C Recommendation 2004 [cited; Available from: http://www.w3.org/TR/owl-ref/.

53. Knublauch, H., Musen, M., and Rector, A. Editing Description Logic Ontologies with the Protégé OWL Plugin. in International Workshop on Description Logics - DL2004. 2004. (Whistler, BC, Canada).

54. Patel-Schneider, P.F., Hayes, P., and Horrocks, I. OWL Web Ontology Language Semantics and Abstract Syntax W3C Recommendation 2004 [cited; Available from: http://www.w3.org/TR/owl-semantics/.

55. Rector, A. Representing Specified Values in OWL: "value partitions" and "value sets". W3C Working Group Note 2005 [cited; Available from: http://www.w3.org/TR/swbp-specified-values/.

56. Motik, B., et al. OWL 2 Web Ontology Language: Profiles. W3C Working Draft 2008 [cited; Available from: http://www.w3.org/TR/owl2-profiles/.

57. Prud'hommeaux, E. and Seaborne, A. SPARQL Query Language for RDF. W3C Recommendation 2008 [cited; Available from: http://www.w3.org/TR/rdf-sparql-query/.

58. Pellet. The Open Source OWL DL Reasoner. 2008 [cited; Available from: http://pellet.owldl.com/.

59. Bechhofer, S. The DIG Description Logic Interface: DIG/1.1. 2003 [cited; Available from: http://dl-web.man.ac.uk/dig/2003/02/interface.pdf.

60. The W3C OWL Working Group. OWL 1.1 Web Ontology Language. Submission request to W3C 2006 [cited; Available from: http://www.w3.org/Submission/2006/10/.