Semantic Tools; Their Use for Knowledge Management in the Public Sector

Stamatios Theocharis and George Tsihrintzis

Abstract In recent years it is globally recorded that there is an increased demand of citizens for greater transparency of operations and accountability of governments to the public. The means to achieve this requirement are the IT technologies through the development of appropriate applications. The reproduction of information has taken massive proportions resulting to the insufficiency of the traditional methods and knowledge management tools. Towards the development of smart applications Internet search and knowledge management, we present in this work related technologies focusing on ontologies.

Keywords Open government data · e-Gov ontology · Reasoning

1 Outlines

In the second part we present critical issues about the opportunities of e-Government through the Semantic Web and the "opening" of public data as an introduction and, in the third part, typical relevant works on the areas of open data and semantic web technologies. In the fourth part we develop the theoretical background on the Semantic knowledge representation using RDF triples and OWL ontologies and fifth place matters on the logical inference and its support languages. In the sixth part we present the concepts and properties that we defined in the ontology developed in the framework of this study while in seventh part we present issues on query submission on ontologies technologies, with relevant examples from our ontology. In the eighth part we present the methodology development and the evaluation of our ontology and in the ninth and final part the conclusions of the relevant study.

S. Theocharis (✉) · G. Tsihrintzis
Department of Informatics, University of Piraeus, Piraeus, Greece
e-mail: stheohar@unipi.gr

G. Tsihrintzis
e-mail: geoatsi@unipi.gr

© Springer-Verlag Berlin Heidelberg 2016

G.A. Tsihrintzis et al. (eds.), *Intelligent Computing Systems*,
Studies in Computational Intelligence 627, DOI 10.1007/978-3-662-49179-9_2

2 Introduction—Presentation of the Field of Interest

2.1 E-Government—The Opportunities Through the Semantic Web

At its early stage, e-government was considered to be synonymous with the ITC introduction in Public Administration, and at a later stage with the development of informative websites of Public Administration and the maintenance of public data in database locally and separately for each public body. Afterwards, upon completing the first four development levels, a fifth one concerning the personalized provision of e-services was added. In this way over the last years e-government has evolved aiming to the integration of smart e-services for the general public as well as the administrative body itself. These services are related to advanced features such as recommendations to traders, advance completion of applications and intelligent information search. The latter level is particularly important if we take into account the number of handled information both electronically and physically and the number and complexity of transactions with the public administration.

The current designation of government websites and online applications are mainly anthropocentric, meaning that most of the content is designed to be accessible by people and not automated by machines. It is based on documents written mainly in the environment of HTML, which is used to describe a structured text with emphasis on visualization and presents limited capabilities in classifying the text sections of a page. So far, computers can efficiently analyze websites in appearance and routines, but generally do not have a reliable way to process the semantics of their content.

In recent years there has been much debate about the evolution of the Internet and the evolution into the so-called semantic web. Initially, the semantic web is expected to contribute to more intelligent access and manage information handled on the internet through new technologies and the development of corresponding new applications. The overall vision of the transition from the existing Internet of static pages into a dynamic network of Web services providers that automatically discover the information sought, negotiate for goods or services that the user intends to purchase or gather information from different sources and unite in homogeneous forms with the ultimate goal of sharing and interaction with other systems based on "common language".

Our interest is focused on the possibilities that can be offered by Semantic Web technologies in the further development of e-government and support open government. The field of e-government, as we believe, may be one of the main consumers of smart services based on the semantic web. This is because the modern trend of open government, is supported by the publication of "open data" and its further connection to with semantic web technologies. By using semantic web technologies, are expected improvements related to both parts of the entire system, i.e. the so-called "front-office" and "back-office":

- As far as the citizens service is concerned: It will greatly facilitate the search of the services provided through the use of government portals (development of semantic portals) and through the physical presence in the service point (in the context of operation of one-stop shop). The search for the appropriate service by the corresponding institution can become more effective and efficient.
- The sharing and searching data among stakeholders in the back-office will be significantly improved by using a common vocabulary and common semantics.

The structural reforms, movements and removal of staff in the public sector has brought great changes and malfunctions in the so-called back-Office of Administration. The reduction of operating costs of public sector is vital and is connected to the efficient functioning of public institutions. This is particularly important if take into account the European economic crisis of recent years. What is required is a corresponding evolution in the overall vision of e-government. Their implementation is associated with the adoption of technologies and methodologies of the semantic web for the benefit of saving resources and infrastructure. Since the technologies of semantic web are not a profitable tool for businesses, essentially one of the key factors for further development of the SW remains the public sector that owns and distributes huge data sizes.

2.2 Public Open Data for the Transition to 'Open Government'

The transformation of governance from the classical model to the model of e-government is now directed towards the adoption of principles and policies of the so-called "open" government. Relative is the initiative "Open Government Partnership-OGP" (and the related guide published by) of a large part of European states as well as the USA regarding the "opening"—sharing the acts of government to citizens with the basic objective of achieving transparency and accountability of the government to its citizens and their participation. In particular, with the concept of transparency, is meant that people know and understand the functions of administration. The controllability of citizens in the operations and the administration effectiveness is associated with the concept of accountability of the administration. With the concept of participation is meant the ability of citizens to form together with the administration the various policies and any procedures that affect them.

By the processing of data provided "open" by the administration to the public, it is possible to produce important information and thus knowledge, in the various areas of interest. The sharing of useful open public data and their processing by interested parties—the scientific community and businesses, must be accompanied with corresponding user rights to its citizens and businesses and can have serious economic and developmental benefits to the states. So there is the public open data, which may concern various cases of public activity, such as financial data, geospatial administrative data, public safety issues, etc.

Regarding Greece the first steps have already been made towards open government with its participation to the OGP initiative (http://www.opengovguide.com/) but also with the adoption of basic principles with a series of acts of low with the "Diavgeia" program (Law 3861/2010) constituting the basis. Under the existing legislation (as amended) the information that can be posted online and constitute open data is the following:

- Laws, acts of legislative content, Presidential decrees
- General regulating acts except for the regulating acts concerning the organization, structure, composition, arrangement, supplies and equipment of the Armed Forces of the country, as well as any other acts, the publication of which causes damage to the country's national defense and security,
- Circulars,
- Economic data on expenditure and state revenue, as budgets, accounts, balance sheets of State stakeholders
- Contracts
- Acts of appointment, acceptance of resignation, transfer, termination of appointment concerning regular employees and employees of special categories, acts of appointment concerning single member bodies and formation of collective management bodies of public institutions
- Acts of appointment concerning Committees and project teams
- Contest announcements and results
- Spatial data relating to the Public land,
- Spatial data related to the local economic and business activity and the residential development
- Administrative division of the country
- Data related to public safety and the fight against crime,
- The distribution of the population
- Social data related to job opportunities, education and health care

This venture in the last five years, has undoubtedly led to the unprecedented expansion of mobile data at both local and the online environment. Moreover, the variety of standards and methods of sharing documents, has created a chaotic situation regarding the recovery of data and their processing. The data are so many, scattered and varied that the real knowledge is extremely difficult to be recovered through the methods and tools of traditional knowledge management used until now. The first steps of this initiative were made without much coherence by those involved resulting in a "Babel" of applications and management methods of the new open data.

In recent years there has been a more systematic effort to more homogeneous and compact data exposure methodology to encourage its use and processing. This methodology uses tools and methodologies of the Semantic Web. This is the basic idea of interest, especially in the case of "Linked Open Data—LOD" in conjunction with the evolution of the Internet to the Semantic Web.

3 Related Work

The evolution of e-government in open government in recent years has been the subject of Administrative Science and especially of Computer Science and directly connected with the open public data as a subset of open data. The requirements and conditions for their disposal, the methodologies for their connection and the development of applications for its processing and further consumption for the production of knowledge, is of particular interest, which is reflected to the number of relevant scientific references in various fields. The relevant references concern the concepts, methodologies and tools of the semantic web, knowledge management and information retrieval, as directly relevant themes with the implementation of related open data.

Theocharis and Tsihrintzis (2014) is the previous work of the authors on ontologies for the support of open public data, which are a special tool for modeling knowledge in the field of e-Government. Ontologies are used as a dictionary for the description and classification of concepts and their correlations in specific fields. In this paper, we had presented the technologies used for the implementation of open data, such as the triplets RDF, the RDF schemas, the OWL language and SPARQL. We had presented the Protégé as a tool for building the ontology and the creation and management of knowledge base around public data. This was considered as a critical point, since the scope of public information is quite complex and the information from different operators using different datasets whereas there is no a commonly accepted dictionary for concepts and relationships between them. Among the critical points we had recorded from the experience of recent years by opening public data in Greece, especially through the "Diavgeia" refers to:

- The existence of a large number of different systems that do not communicate with each other since they do not support the existing interoperability standards.
- Many bodies store the same data in different places, ignoring the existence of duplicates, resulting in waste of resources and creating conflicts in their use
- Depending on their structure the bodies use similar—but not exactly the same terminology for naming organizational units having the same object.
- There is confusion as to the correct completion of the thematic fields and the type of decision.
- The search for a published decision in the current search system, is based on the word or key expression either by body or to all the stakeholders but the results are not related.

We have been able to conclude that there is a question of efficiency and overall reliability of the information search system which resulted in delays and depreciation of the system of open data especially from the public information producers themselves.

In this paper we did not fully develop the overall model of public administration, but only the part concerning the description of so-called administrative acts and the concepts associated with them. For the demonstration of the ontology constructed

in this work, we considered a real scenario of administrative act concerning the municipal elections. Through this, we introduced the relevant concepts and their relationships. By using the reasoners offered by the Protégé we presented the evaluation process of the ontology on the consistency and search of any logical errors. Also we had presented the two main tools available in the Protégé 4.2 to query the knowledge base that we have created and the results of the submission of relevant questions.

By the development of this ontology we recorded significant findings for the construction of the overall description model of public administration in order to be led to the correct connection of the already open public data. One of the main problems we faced in this work involved the compulsory transfer in English of Greek Public Administration terms as there is no semantic content management software in Greek language oriented to public administration.

In Heath and Bizer (2011) we provided general principles for the creation of linked open data on the Web, and we give an overview of the technologies of the Web. These principles include the use of URI's and the connection among them as well as the use of RDF and SparqL in search for information. The application of the overall architecture of WWW, which includes the HTTP protocol as a generalized mechanism of accessibility and HTML as a general formatting tool for sharing structured data in a specified format worldwide, is presented as a general principle. Reference is also made to two key issues: the publication and consumption of linked open data by presenting basic methods for connecting information that could serve as linked open data. Also presented are standards for connection of open data and methodologies for their implementation. Finally are presented, the applications and architectures on which they are build, so that it will be able to constitute a training guide for researchers involved in developing applications in open data.

In (Educational Curriculum for the usage of Linked Data Project, in http://www.euclid-project.eu/), an educational tool through a European program for the consolidation of related data and methods for their implementation to interested professionals or researchers, is presented. In particular, one of the main objectives of the project is to support different teaching scenarios. Therefore, EUCLID provides an extensive training curriculum, covering the main technologies, tools, use cases and skills that need to be acquired in order to complete both basic, as well as more complex, tasks related to dealing with Linked Data. Among the topic that are covered are: the basics Linked Data principles, the SparqL as a Query tool, the visualization and the search in the linked data and the creation of Linked Data applications. In addition, one of the main features of EUCLID's curriculum is that it not only covers the six core topics related to learning how to use and employ Linked Data through the six developed modules. It also provides a set of diverse materials supporting a variety of learning scenarios.

The page in (http://www.w3.org/standards/techs/gld#w3c_all), summarizes the relationships among specifications, whether they are finished standards or drafts in the domain of Government Linked Data. Each title in this page links to the most recent version of a specific document, such as the RDF Data Cube Vocabulary, Data Catalog Vocabulary etc. It's about the W3C recommendations in the domain

of Semantic Web and the Open Data. Through the various links the ability to examine one concrete examples is provided in addition to prototyping in specific themes. For instance, the document in (http://www.w3.org/TR/2014/REC-vocab-org-20140116/) describes a core ontology for organizational structures, aimed at supporting linked data publishing of organizational information across a number of domains. It is designed to allow domain-specific extensions to add classification of organizations and roles, as well as extensions to support neighbouring information such as organizational activities.

There are also several reports for associated open data among which we refer Parundekar et al. (2010) which presents an approach to the production of equivalent relations through research in the environment stating the equivalent classes in various entities. In addition in Jain et al. (2010) we refer to the ontologies behind the Linked Open Data and present a system about finding schema-level links between LOD datasets. A view on the ontology of public administration is given in Goudos et al. (2007) which will form, according to the authors, the cognitive basis for developing semantic web applications in the field of e-Government.

4 Semantic Representation of Knowledge

4.1 The RDF Data Model

The Resource Description Framework—RDF is a W3C standard for simple description of metadata on the internet although there are many who consider it as an ontological language. This model enables you to implement graphs of data and to share between users and machines. For this reason various programming languages such as JAVA and Python, support through appropriate libraries the RDF data. Also many software applications have been developed for the use of such data and the production of these knowledge.

The basic concept of the representation of knowledge is the resource. In this sense, we mean any entity the World Wide Web such as a website, a part or a group of web pages, electronic files or even objects that are not immediately available online, for example a book. The description of the resources of the Internet with the RDF focuses primarily on efficiency of metadata such as title, author's name, creation date, etc. The general idea is that each resource has one or more properties which have specific rates or they may be themselves resources from the Internet. To summarize, the RDF information is given as suggestions in the form of triples subject—properties—object. In this triple, the subject is a resource on the Internet which is identified by a specific unique URI. The property attributed to the subject is specifically described in another resource of the internet (and therefore is itself a resource). The object can either be a literal or is itself another resource. A set of RDF triples can be perceived as a graph. In this graph the objects and subjects play the role of nodes and properties play the role of the connecting edges.

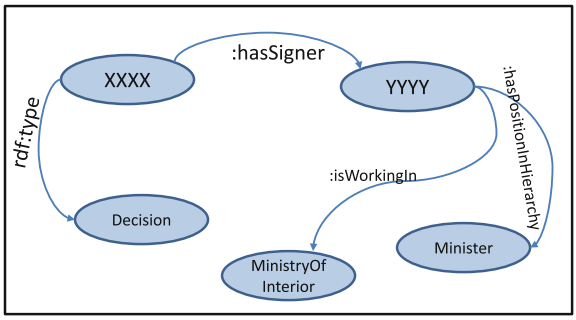


Fig. 1 An RDF graph describing a statement in e-government domain

For example in the model of public administration that we have developed in this paper, we consider the case of the "*XXXX decision is signatory to YYYY who is the Minister of the Interior*". The RDF graph that describes the above statement is shown in Fig. 1.

The syntax used in the RDF representation of the data, is based on the XML standard referred to as RDF/XML syntax. For reasons of generality the RDF uses URI references to identify the entities that are in the position of the subject the property and the object.

4.2 The URI's Use

In order to implement the RDF representation, where basic concept is the resource we use as unique identifiers the Universal Resource Identifiers—URIs. The known URLs (Universal Resource Locators) or the strings that we use to identify each site on the internet is a subset of URIs. The URIs generalize logic of URLs in the sense that anything that can be described and recovered can be identified in a manner analogous to websites. The URI that is used in an RDF statement is characterized as URI reference—URIRef. A URIref comprises a URI and an optional fragment identifier. For example the URIRef: http://www.example.org/index.html#doc1 consists of the URI http://www.example.org/index.html and the fragment identifier *doc1* which distinguish from the URI of the use the # symbol. It is used for the first part of the URI to be entrusted to a relative prefix which is then accessible. In this way each person can identify the resources they wish to publish electronically uniquely worldwide. So for example, a ministry can use for one of its resources the URI: http://www.ministryofinterior.org#path1/docl, which is a component of a unique URL assigned to the institution (http://www.ministryofinterior.org#) followed by the corresponding internal path (/*Path1*/*doc1*) with which each resource is identified in the internal system featured. With RDF statement: "*PREFIX mint:* <http://www.ministryofinterior.org#>", the URI could be accessible in the form of mint: *Path1*/*doc1*. Because the URI are used in the manner mentioned as unique identifiers of resources worldwide they are considered as strong identifiers.

4.3 RDF Schema Specification Language

To RDF data model does not offer the possibility of restrictions on reporting relationships that exist between the properties and resources. For example in RDF there is no statement to define that "*signatory of a decision can only be a minister*" or that "*the MinistryOfInterior is the individual of the class Ministry*". So apart from the RDF descriptions of resources, the RDF data model is also supported by a schema definition language (sets of classes and properties), the RDF Schema Specification Language (RDFS). With the help of RDFS, it is possible to identify mechanisms for determining classes of resources and the limitation of possible relationships between classes using appropriate correlations. An RDF schema consists of statements classes, attributes and relationships between classes. Similar resources are grouped under the same heading. Based on the above, we can distinguish three different levels of abstraction in the data model RDFS: In the lower level there are the same resources (documents, websites, persons, or whatever).

The next level of abstraction is the data level, which is the description of information resources using specific vocabularies, which are described in the last level, the shape level. The shape level is the abstraction level where RDF shapes are developed to facilitate semantic resource description. At this level, the classes represent abstract entities and collectively refer to sets of similar objects. For the graphical display of RDF/S descriptions and figures a directed graph model is used, with labels both to the edges and nodes which can easily combine many different vocabularies and be expanded simply by adding more edges. The nodes represent objects (resources or classes) and edges represent relationships between nodes (properties). As nodes we represent also Literal types, i.e. strings and other basic data types as defined in the standard XML Schema. The nodes are represented as gaps and individual values as parallelograms. The edges can be of three types: Performance attributes, creating instances and employees. The performance attribute edges represent attributes of nodes and relations between them while the subsumption edges used to indicate a shape layer that a node (class) or property, are a subclass of a broader semantic node or property respectively. Finally, the snapshots creating edges form the link between RDF standards and RDFS, allowing the creation of instances of a class and assigning types to information resources described.

For better sharing of resources RDF descriptions and schemas on the web the model RDF "borrows" same syntax as XML. This leads to the RDF/XML syntax of RDF schemas and RDF description of resources shown in the graph representation.

4.4 Web Ontology Language—OWL

OWL was created to satisfy the need for a web ontology language and is one of the W3C's recommendations on implementing Semantic Web technologies (http://www.w3.org/OWL/). OWL is designed to be able to process the content of

information. The purpose of OWL (Uschold and Gruninger 1996) is to provide a standard format that is compatible with the Internet architecture and the Semantic Web. The standardization of ontologies in OWL language will make the data on the Web more machine processable and reusable applications. Thus, scalability, convertibility and interoperability possess high priority in language design. Compared with existing technologies, OWL exceed the known and widely used languages XML, XML Schema, RDF and RDF Schema as it supports greater clarification of the Web content than machines, providing additional vocabulary along with a formal semantics.

The OWL adds larger vocabulary for the description of properties and classes such as relations between classes (e.g. disjointness), the number of elements in a set (cardinality), logical combinations of classes (union, intersection), equality, richer typing of properties, characteristics of properties (e.g. symmetry, uniqueness, transitivity), and enumerated classes (enumeration).

An advantage of OWL ontologies is the availability of tools which may draw conclusions about these (reasoners). These tools produce general support related to stated rules or events that follow specific morphology and vocabulary. OWL is developed as an extension of RDF vocabulary. It derives from the web ontology language DAML+OIL and it is written in XML format that can easily be independent of the operating system and language implementation of a computer. An OWL ontology includes descriptions of classes, properties and their instances. Taking into account such an ontology, the formal semantics OWL specifies how to produce the logical conclusions, i.e. facts not literally present in the ontology, but implied by the semantics. These implications can rely on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

As the Semantic Web is distributed, the OWL should allow to gather information from distributed sources. This is done partly by allowing ontologies to be associated, including clearly the import of information from other ontologies. The practical applications of OWL, include Web portals, which can be used to create the classification rules, in order to improve the search, the Multimedia Collections, which can be used to enable content-based searches of media and Web Services, where it can be used for the discovery and composition of web Services as well as for rights management and access control.

Table 1 shows the correspondence between the abstract syntax of OWL class descriptions and descriptions of concepts of descriptive logic. As we can see, in this match, derives the semantics of the structural elements of the OWL language. A similar correlation can be set in the case of class axioms of OWL language. These axioms may be assigned to axioms of affiliation and equivalency between concepts of a Description Logic. The correlations are shown in Table 2 and in most cases are quite obvious. The only case that is worth comment on is the one of foreign classes. From the set theory we know that two sets are disjoint if their intersection is the empty set. If transferred from the field of sets to the abstract space of concepts then we can define two concepts as foreign if the intersection of their falls into the empty concept. It is pertinent to note that there is an alternative statement that two

Table 1 Correspondence between the abstract syntax of OWL class descriptions and descriptions of concepts of descriptive logic

OWL syntax	DL	Example	Semantic annotation
IntersectionOf	C∩D	CivilServant∩CitizenAttributes	All the Civil Servants that have specific attributes
UnionOf	C∪D	CivilServant∪Professor	Anyone that is civil servant or a professor
Atomic megation	¬C	¬CivilServant	Anyone than is not a Civil Servant
hasValue(o)	∃R.{o}	∃hasSigner.CivilServant	The signer of a document must have value CivilServant
allValuesFrom(C)	∀R.C	∀CivilServant.{isWorkingIn some State}	Civil servant is anyone that is working in some unit of State

Table 2 Class axioms in OWL

OWL syntax	DL
Class(A partial C1,…Cn)	A ⊆ C1∩…∩Cn
Class(A complete C1,…Cn)	A ≡ C1∩…∩Cn
EnumeratedClass(A O1,…On)	A ≡ {O1}∪…∪{On}
SubClassOf(C1, C2)	C1⊆C2
EquivalentClasses(C1,…Cn)	C1≡…≡Cn
DisjointClasses(C1,…Cn)	Ci∩Cj⊆⊥

concepts are disjoint. More specifically the term C is foreign to the meaning D if-f C⊑¬D.

Table 3 shows the correlation between the properties of the OWL axioms and axioms of roles descriptive sense. Once again we observe that OWL does not offer anything more than the expressive ability that gives us a Description Logic System even where this is not immediately obvious, as in the case of functional roles explained above. From this table however we observe something interesting. The axioms of definition of range and domain of a property, which we met in language RDF-S, may themselves be encoded by using Descriptive Logic axioms. More specifically a form axiom definition of the range of ObjectProperty (R domain (C)) corresponds to the office of entry for ∃R.⊤⊑C, while an axiom definition of the form of range ObjectProperty (R range (C)) corresponding to the entry office, ⊤⊑∀R.C. Intuitively the first axiom tells us: "*If an object is connected through the relation R to*

Table 3 Property axioms in OWL

OWL syntax	DL
SubPropertyOf(R1,R2)	R1⊆R2
EquivalentProperties(R1,R2)	R1≡R2
ObjectProperty (R domain (C))	∃R.⊤⊑C
ObjectProperty (R range (C))	⊤⊑∀R.C

something, *then this object is of type C*". So this axiom tells us that all individuals who are in the first position of a pair (a, b) which belongs to the relation R is type C. The second axiom tells us that "*for all objects of the world*, *whenever they are connected to another object* via *the relationship R then the object is of type C*". So all individuals that are in the second position of a pair (a, b) which belongs to the link.

5 Reasoning Tools

The acquisition of knowledge from a database or from a base of knowledge is often associated with the submission of relevant questions through appropriate interfaces. The answer to a question in semantic web, is not just about data control (such as databases), but it requires the implementation of complex considerations that cover a huge crowd classes and instances. Classes and instances are linked together under declared rules and constraints that describe specific events and roles. For this reason, the combination of declared facts and concepts in an ontology are particularly important so that the correctness or not of another event occurs as a conclusion or so that a new unexpected fact is produced. This process is described as reasoning and is one of the main differences between databases and knowledge bases of the semantic web. Of particular interest is the sub-process of reasoning concerning the production-addition of new facts in ontology referred also as inferencing.

The reasoning depends on the way of representing the events in the ontology. This representation is formulated by using formal languages based on various, such as predicate logic and descriptive logic. The last, are used mainly for supporting the semantic web and based on Open World Assumption. In descriptive logic, we use the concepts, objects and people in order to define the nomenclature relating to the field of interest. A typical architecture of a system is given below (Fig. 2) based on descriptive logic.

The knowledge base is the combination of statements of two ingredients: (a) Terminology Box—TBox comprising the vocabulary of the field of interest

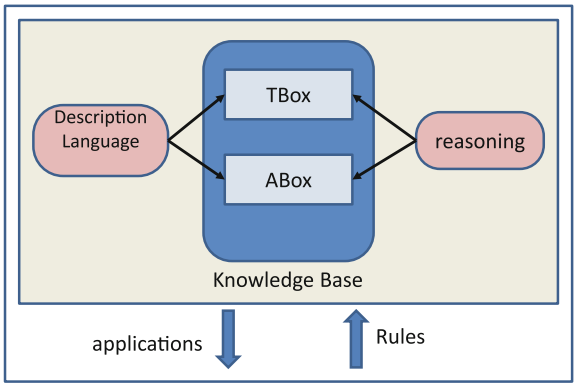


Fig. 2 A typical architecture of a reasoning system

through the concepts and definitions of properties and (b) Assertion Box—Abox comprising all axioms/allegations on individuals included in TBox.

As a vocabulary we mean all the concepts (which usually group together individuals) and the roles referring to binary relations between individuals. The concept and roles description languages for the knowledge base, are based on a basic language, Attributive Language, that provides key manufacturers with common syntax rules.

Special software has been developed under these logic (for example OWL2 Query Tab of Protégé presented below) which support the operation of ontologies management applications by using reasoners such as Pellet, Fact ++ and HermiT used in Protégé. The reasoning (reasoning) provided by the software are particularly useful service and mainly include the following categories of controls: (a) knowledge base consistency checking, (b) concepts classification and (c) instance checking, with the sense of the calculation of classes, which include snapshots of the database.

5.1 SWRL Rules

The SWRL (Semantic Web Rule Language) is a language of the Semantic Web standards and is based on a combination of OWL DL and OWL Lite sublanguage of OWL with Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. This proposal extends the set of OWL axioms by introducing Horn-like rules (O'Connor 2009). In this way it allows a combination of Horn-like rules with an OWL database. At the same time, it provides a high-level abstract syntax for Horn-like rules, which are expressed in terms of the OWL language (classes, properties, individuals). The rules are in the form of contributory fault between an antecedent and a consequent part. The antecedent part is the main body of a rule and it is known as body, while the consequent part is the consequence of the rule and is known as head. The point at which a rule is intended to be interpreted as follows: whenever applicable conditions laid down in the antecedent of a rule, you should apply the conditions laid down in the consequent part of the rule. The rules are stored as part of the ontology and also it is possible that they interact with reasoners. While the provision of reasoning tools that support SWRL constantly growing, we can distinguish Bossam, R2ML, Hoolet, Pellet, KAON2, RacerPro and SWRLTab. It is known that carries OWL inference capabilities through the features of OWL properties, such as inversion, symmetry and composition of indirect relationships through a combination of direct relations (transitive property). The SWRL carries inference capabilities through the rules. To avoid repetition necessity between OWL extrapolations and SWRL extrapolations would be desirable that rules engines could follow the OWL characterizations. This means that the OWL features should be translated to SWRL equivalents. For this reason, in SWRL is perfectly possible to define rules for attributes that represent symmetry, inversion and transitive possibility.

5.2 The Query Language SQWRL

The library of Protégé-OWL contains a number of built-ins that extend the SWRL in SQWRL. The SQWRL (Semantic Query-Enhanced Web Rule Language) is a language based on SWRL standard for performing queries in OWL ontologies. It has similar methods to SQL to retrieve knowledge from OWL. The determination of SQWRL is done using a library of SWRL built-ins that structure an efficient query language over the SWRL. The built-ins contained in this library are determined by the SQWRL Ontology. The common prefix is sqwrl. The Jess Rule Engine is necessary for the performance of SQWRL questions. The SQWRL questions being built on SWRL can be used for retrieving knowledge which has deduced from SWRL rules. Moreover, SQWRL questions are free to cooperate with other built-in libraries. This free choice and use of built-ins to the questions, provides a continuous expansion means of the expressive power of the query language.

6 Presentation of Our Ontology Through Protégé

6.1 The Ontology Development in Protégé 4.3

Further to our previous work in Theocharis and Tsihrintzis (2014), in this paper, we reviewed some of the structures of the initial model of public administration and added new concepts, properties and instances. The aim of this study was to adapt the ontology so that it describes the actual image recorded today in the public service and not a theoretical model of knowledge. So the concepts and properties that we defined correspond to reality while many of the instances are images of the actual structure of public administration.

In this study we used version 4.3 of Protégé, as one of the most updated, oriented OWL ontologies. As a big advantage of these versions 4.x., we highlighted the friendliness towards the creation of the properties and their limitations on the classes and instances. On the other hand, the concealment of SWRL and the replacement of the corresponding tab that the versions 3.x offered with rules tab, restricts the possibilities for developing complex rules on classes or instances.

6.2 The E-Government Ontology

6.2.1 Defining Classes

Like the previous version of our ontology, the present edition also describes the interaction environment of public administration as a system under the tripolar model comprising three main pillars: the state, citizens and businesses. In our

ontology, this is indicated by the definition of the class *Trader* which has as subclasses the: *Citizen*, *Enterprize*, *State*, and an additional two (excluding basic pillars): *NGO* (*Non Governmental Organizations*) and *EuropeanUnion*. The case of subclasses of State interests us particularly, since each country has its own state structure but also internally in each country there are different views as to what is considered as state. Here, the class *State* depicts the Greek reality as it emerges from relevant legal texts e.g. the *Constitution* of our country. By this logic structure created shown in Fig. 3.

We also considered citizens as a member of the system to the same class. Fundamental also is the character of the profession which gives us a way to distinguish citizens. So the *civil servants*, *judicial officials*, *professionals*, *pensioners*, *politicians*, *private employees*, *teachers*, *students*, *the unemployed*, are represented through their respective subclasses of Class *Citizen*.

To describe the properties of the citizens we defined the class *CitizenAttributes* with subclasses the *Gender*, *IllegalActivity*, *Married*. Class *Gender* has two disjoint individuals (*man*, *woman*). The class *IllegalActivity* has additional subclasses according to the type specified by the statements in the General Class Axioms, shown in Fig. 4.

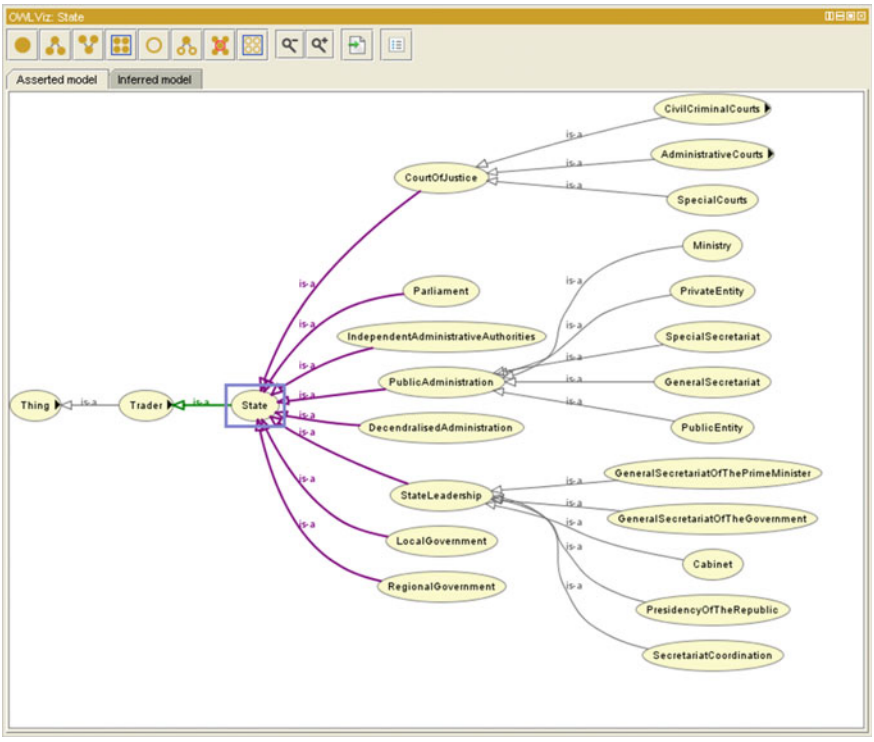


Fig. 3 The state class and subclasses in OWLViz tool of Protégé

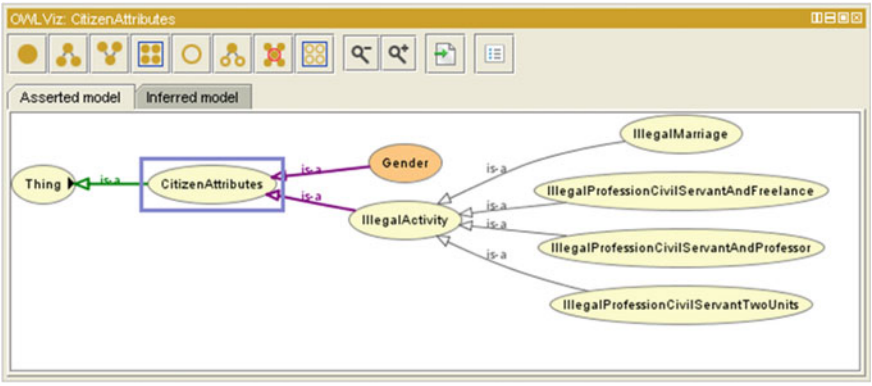


Fig. 4 The CitizenAttributes class and subclasses in OWLViz tool of Protégé

This model supports the so-called "*separation of powers*" of the Constitution into *executive*, *judicial and legislative*. These concepts are provided by *StateAuthorities* class and subclasses *Executive*, *Judicial* and *StateLegislation* respectively as shown in Fig. 5. At the same basic level with the class *Trader* and *StateAuthorities*, we defined the class *Services* to describe the categorization of the services provided by the state to citizens (subclass *G2C*), to businesses (subclass *G2E*) and to the state itself (*G2G*).

From another perspective, the services and procedures also cover the so-called front-Office and back-Office of Administration. The visual is covered by the class *Process* and respective subclasses that appear in Fig. 6.

In this ontology, we chose to describe the properties of organisms as a class *OrganizationAttibutes* to include two main characteristics of agencies as subclasses: the *bureaucracy* (in the sense of hierarchical organization) and *organizational structure* (in the sense of the units which constitute organizations).

For a description of the operations of the state, we defined the class *ActsAttributes* with subclasses *Gradation*, *Kind*, *Priority*, *Topic* for describing their essential characteristics. In parallel, we have provided a description of the transaction points to the Public Administration with the class *TransactionSpot*. Transaction points can be understood either by their physical nature (*AgentOffice*) or their electronic form (*DiavgeiaSpot*, *WebSite*). The class *Location* and the subclasses *City*, *Prefecture*, *Region* implement the requirement that transaction points are located within a specific spot.

6.2.2 Defining Properties

The protégé enables definition of object properties and datatype properties. The object properties have the role of correlations among individuals and practically implement the triplets RDF. The object properties that we stated in our ontology,

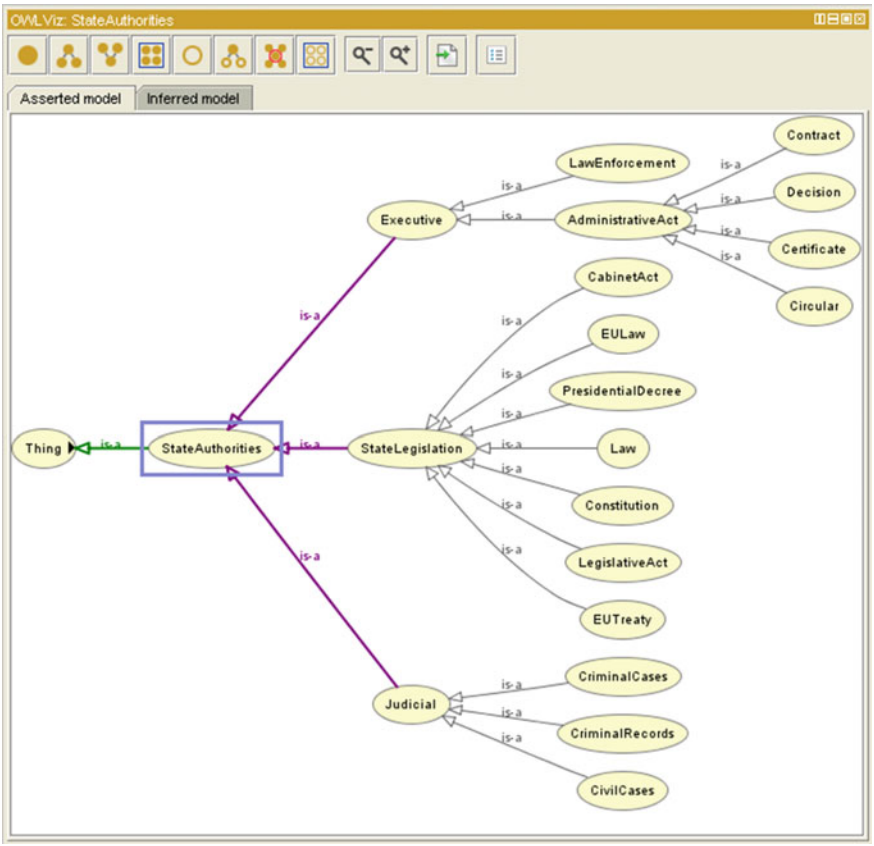


Fig. 5 The StateAuthorities class and subclasses in OWLViz tool of Protégé

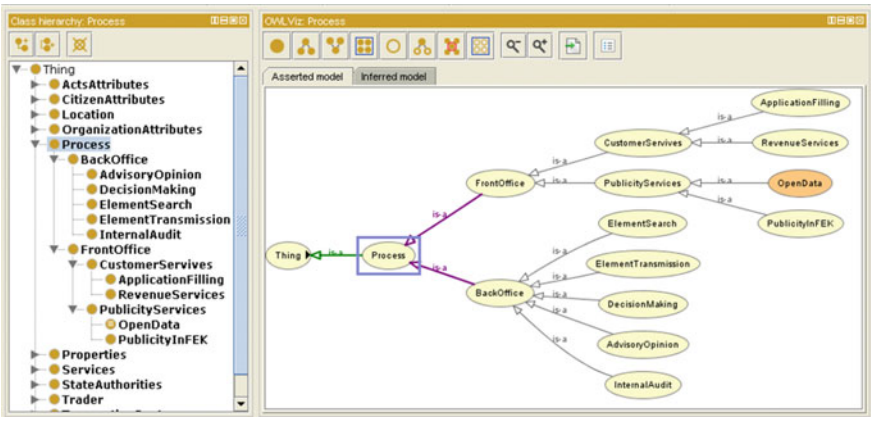


Fig. 6 The process class and subclasses in class hierarchy view and OWLViz tool of Protégé

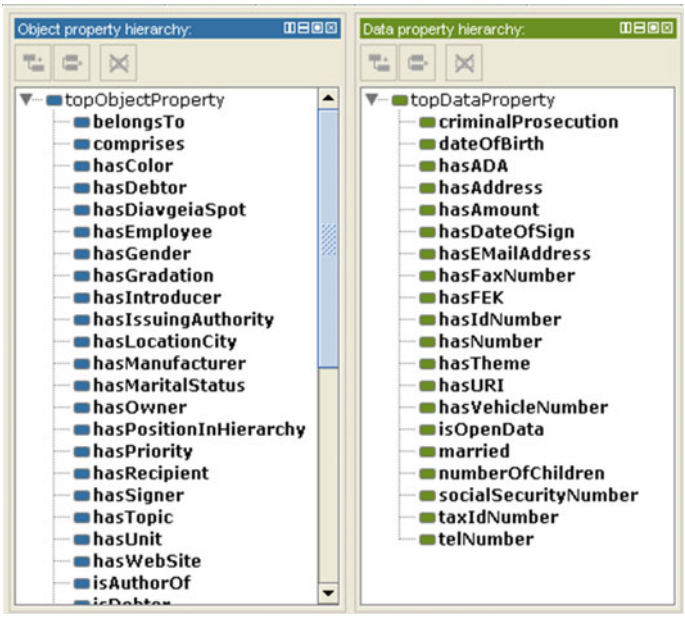


Fig. 7 The ontology properties in hierarchy view of Protégé

are shown in Fig. 7. We note as particularly important the *belongsTo* which is transitional and the inverse of the comprises which we use to attribute semantically that "*a City belongs to a Prefecture*" or "*a Prefecture comprises a Region*". Also for the corresponding requirement that "*a Prefecture belongs to a Region*" or "*a Region comprises a Prefecture*". Also important is the *hasPositionInHierarchy* which has the union of the domain classes *CivilServant* and *PrivateSectorEmployee* and range class *Bureaucracy*. This property is used to describe the requirement that "*an Employee has a specific position in the bureaucracy of the organization*". We also note the *hasSigner* (the inverse of the *isSignerOf*) and *hasIntroduser* (the inverse of the *isIntroduserOf*) properties. These properties have as a subject the acts adopted by the executive authority (*domain* = *Executive*) and as an object the civil servants (*range* = *CivilServant*). With these qualities we implement the description of the requirement that "*an Executive act has signer a civil servant*" or "*an Executive act has introducer a civil servant*". By the property *hasIssuingAuthority* we describe the requirement that every transaction (of the *Executive* or of the *Legislative authority*) has a competent authority which issued it. This is achieved with the statement as a domain of the association of classes *Executive* and *StateLegislation*.

The datatype properties we have set are shown in Fig. 7. The properties of this kind are used to connect the logic of RDF of certain individual with specific literal. For example, every legislative act or some of the administrative acts shall be published in the daily newspaper of the Government and take a unique number (FEK).

This requirement is indicated by the use of the property *hasFEK* which has as a domain the union of *AdministrativeAct*, *CabinetAct*, *Decision*, *Law*, *LegislativeAct*, *PresidentialDecree* and as a range type: string. Also, according to the Clarity program, specific categories of administrative acts and instruments are published on the internet, making the so-called open public data. Any document notified on the specific website (www.diavgeia.gov.gr) is characterized by a unique number ADA. This requirement is described in the property *hasADA* which has the same domain and range with the capacity *hasFEK*. Also by using the property *isOpenData* which is type Boolean, we describe the fact that "*a particular kind of the administrative acts or Laws are Open Data*". This is been achieved by the definition of the general class axiom: "*hasADA some string SubclassOf isOpenData value true*".

6.3 The Use of RDF, RDFS, OWL and SWRL Through a Case Study

The RDFS statements involving domain-range are particularly important. This is because the declaration of a class as a domain of a property implements the part of the RDF triplet concerning the subject. In other words, the individual having *rdf:type* the domain of a property is the subject and the particular property is the predicate of a RDF triplet. Correspondingly, the individual having *rdf:type* the range of the property can be the object of the triplet.

For example the property *hasSigner* we have stated to have the domain and range Class *Executive* Class *CivilServant* as shown in the following code section RDF/XML.

```
<owl:ObjectProperty rdf:about="&pa;hasSigner">
  <rdfs:range rdf:resource="&pa;CivilServant"/>
  <rdfs:domain rdf:resource="&pa;Executive"/>
</owl:ObjectProperty>
```

As shown, the *hasSigner* resource is *owl:ObjectProperty* while *CivilServant* and *Executive* resources are *rdfs:range* and *rdf:domain* respectively. In this way we implement the triplet that performs semantic requirement that "*all acts of the executive authority has as signatory a civil servant*."

We should note at this point that the domain and range do not function as limitations, such as the domain of a mathematical function, but function as a classifier for specific individual. For instance with the statement:

```
<owl:NamedIndividual rdf:about="&pa;Ind_Test1">
  <hasSigner rdf:resource="&pa;Citizen5"/>
</owl:NamedIndividual>
```

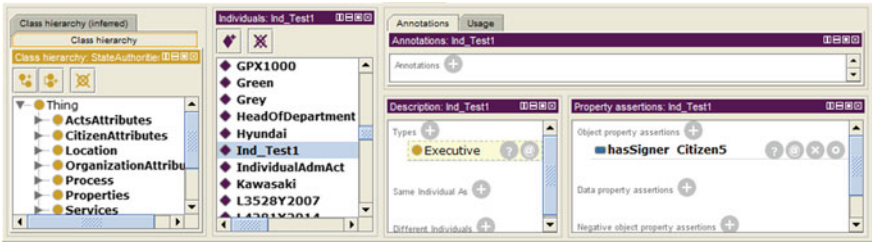


Fig. 8 The reasoner inference about the classification of Ind_Test1 individual

is implied that *Int_Test1* the subject is associated with the object *Citizen5* with the property *hasSigner*. We have stated that that "*Ind_Test1 has Signer Citizen1*". As shown in Fig. 8 using the reasoner's protégé, it is concluded that the *Int_Test1* is type *Executive*, which has not been declared and we do not know if it is a correct conclusion for the classification concerned.

Also important is the function of the features of the properties. Such features are "functional", "transitive", "symmetric" etc. Their use requires attention and understanding of their operation because although they are very useful, they can easily lead to logic errors.

For example, when a property is classified as functional, this means it can get at most one value. That means that the subject in this property can be associated with at most one object. This feature does not work limited as to the subject, but conclusively. For example, if you declare that:

```
<owl:NamedIndividual rdf:about="&pa;Citizen1">
    <hasPositionInHierarchy rdf:resource="&pa;Introduser"/>
    <hasPositionInHierarchy rdf:resource="&pa;Rapporteur"/>
</owl:NamedIndividual>
```

i.e. if we declare that "*Citizen1 has Position In Hierarchy*" both *Introduser* and *Rapporteur*, (with the use of functional property *hasPositionInHierarchy*), then the reasoner of Protégé concludes that individual *Introduser* and *Rapporteur* is identical, as shown in Fig. 9.

For a highest level of expressivity we can use OWL. Relation between classes can be formally modelled based on description logics (mathematical theory). Because OWL relies heavily on the reasoner, it is possible to express complex constructs such as chained properties for instance or restriction between classes. OWL serves to build ontologies or schema on the top of RDF datasets. As OWL can be serialised as RDF/XML, it is theoretically possible to query it via SPARQL, yet it is much more intuitive to query an OWL ontology with a DL query (which is usually a standard OWL class expression). An example of OWL constructs serialised in RDF/XML is given throughout the object Property *hasPositionInHierarchy*.

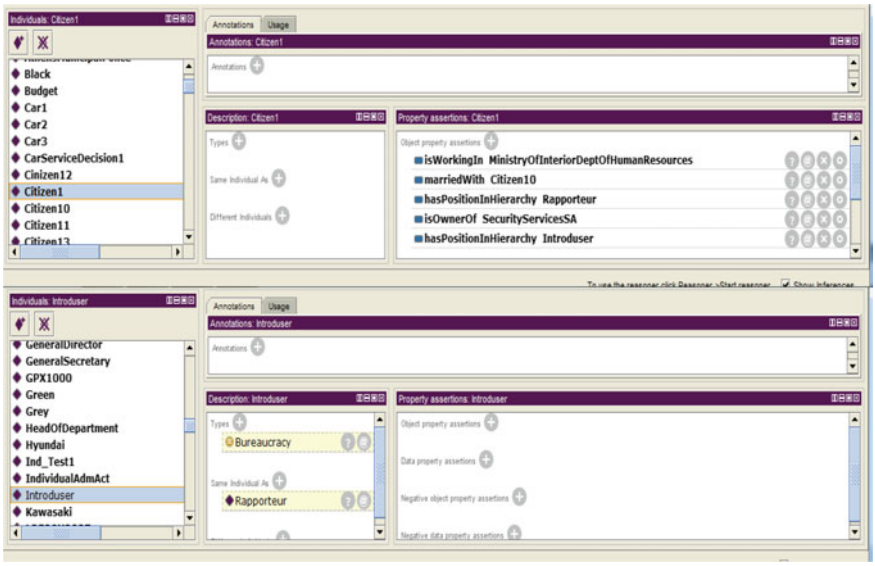


Fig. 9 Classification of Citizen1

The *hasPositionInHierarchy* property connects the *Citizen* classes and *PrivateSectorEmployee* with the *Bureaucracy* which has as domain the union of the first two and the third range from those classes. As shown in the following part of the RDF/XML code, the statements of the domain and range follow the standard RDFS and the concept of classes and their union are attributed with the use of OWL.

```
<owl:ObjectProperty rdf:about="&pa;hasPositionInHierarchy">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdfs:range rdf:resource="&pa;Bureaucracy"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&pa;CivilServant"/>
        <rdf:Description rdf:about="&pa;PrivateSectorEmployee"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
```

A more complex requirement is presented in the following example. One of the requirements of the Civil Servants Code specifies among others that: "*1. Civil*

servant is any citizen who works in a public body. 2. Each civil servant holding a position in the bureaucracy of the organization, including: *Rapporteur*, *Head of Department*, *Director*, *General Director. 3. A civil servant cannot possess two positions in two public bodies.*"

The requirement 1. is described in the declarations:

```
<owl:Class rdf:about="&pa;CivilServant">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&pa;isWorkingIn"/>
        <owl:someValuesFrom rdf:resource="&pa;State"/>
      </owl:Restriction>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&pa;Citizen"/>
  </owl:Class>
```

The statement describing the bureaucracy of an organization that is necessary for requirement 2. is enumeration. This is done by the OWL statement *<owl: equivalentClass>* together with the statement of the members made with the statement *owl:oneof rdf: parseType*='*Collection*' *and RDF* statements of the form *rdf:Description rdf:about=NamedIntividual* to give the respective individual.

```
<owl:Class rdf:about="&pa;Bureaucracy">
    <owl:equivalentClass>
      <owl:Class>
        <owl:oneOf rdf:parseType="Collection">
          <rdf:Description rdf:about="&pa;Director"/>
          <rdf:Description rdf:about="&pa;GeneralDirector"/>
          <rdf:Description rdf:about="&pa;Rapporteur"/>
          <rdf:Description rdf:about="&pa;GeneralSecretary"/>
          <rdf:Description rdf:about="&pa;HeadOfDepartment"/>
          <rdf:Description rdf:about="&pa;Minister"/>
        </owl:oneOf>
      </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&pa;OrganizationAttributes"/>
  </owl:Class>
```

The requirement 3. is described with the class definition *IllegalProfession CivilServantTwoUnits*, with the condition that an individual in order to belong to the class must meet the status *isWorkingIn* at least twice, as seen in the following code fragment:

```
<owl:Class>
    <rdfs:subClassOf
rdf:resource="&pa;IllegalProfessionCivilServantTwoUnits"/>
    <owl:intersectionOf rdf:parseType="Collection">
      <rdf:Description rdf:about="&pa;Citizen"/>
      <owl:Restriction>
        <owl:onProperty rdf:resource="&pa;isWorkingIn"/>
        <owl:onClass rdf:resource="&pa;State"/>
        <owl:minQualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">2</owl:minQualifiedCardinality>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

7 Data Mining Technology from Ontologies

As mentioned above, the knowledge held by experts in each field, can be described based on a common vocabulary with the help of ontologies. Many times, however, the knowledge derived from Data mining tools and the knowledge of experts contained in a ontology is often written in different knowledge representation models or in different shapes. So the first step is the representation of knowledge in a common format using translation mechanisms from different formats to a common one. In the next step we confront issues of resolving any disputes and then processing them.

7.1 SPARQL

Knowledge in Semantic Web is in the form of triplets RDF, to which specific languages are focused. Their main object is the recovery of results upon submission of tailored questions in the form of triplets. Such languages are RQL and SPARQL. With these languages ontological knowledge that has been represented in RDF can easily be questioned, cut, transformed or concentrated. This knowledge often stems from a Web Service and is updated either from a specialist, or it is the result of a Data mining process. In our work we are interested in SPARQL language, which we can use in different ontology management tools such as Protégé.

The SPARQL (SPARQL Protocol and RDF Query Language) is the query language for RDF documents which has become a prototype for the W3C in January 2008 (www.w3c.org). The SPARQL is based on previous query languages for RDF, like RDQL and SeRQL with which they form a family of query languages. Their use depends on the applications developed and run every time. One

of their common features is that they comprehend the RDF data as simple triples without some shape or another ontological information unless it is explicitly set to the RDF document. Also, these languages are all SQL-like, i.e. mimic the SQL syntax styles. The basic building block of SPARQL is the triple pattern and is an individual (elementary) question. A standard triple is actually an RDF triple (subject—predicate—object) which may contain a variable in one or more of the three positions. A prerequisite for the question to give a result, in other words for the variable to get some price is the existence of pairing with some RDF triple of the document upon which the question is put. A typical example is the question of the form:

SELECT ?subject ?predicate ?object
WHERE {?subject ?predicate ?object}

which returns all RDF triples contained in the document.

Remarks by the implementation of SPARQL query

The Protégé environment allows the submission of SPARQL queries. In this environment we submitted specific questions shown in Fig. 10.

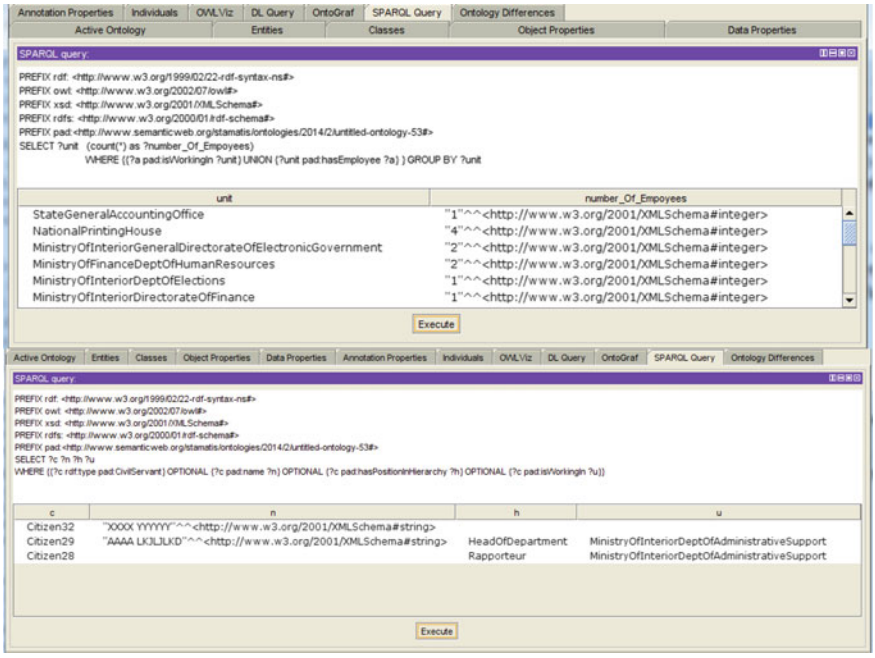


Fig. 10 a and b queries and their results in SPARQL Query Tab of Protégé

a. Number of employees per organization unit:

```
PREFIX pad:<http://www.semanticweb.org/stamatis/ontologies/2014/2/untitled-ontology-53#>
SELECT ?unit (count(*) as ?number_Of_Empoyees)
        WHERE {{?a pad:isWorkingIn ?unit} UNION {?unit pad:hasEmployee ?a} } GROUP BY ?unit
```

b. The data about employees in the units of organizations:

```
PREFIX pad:<http://www.semanticweb.org/stamatis/ontologies/2014/2/untitled-ontology-53#>
SELECT ?c ?n ?h ?u
WHERE {{?c rdf:type pad:CivilServant} OPTIONAL {?c pad:name ?n} OPTIONAL {?c pad:hasPositionInHierarchy ?h} OPTIONAL {?c pad:isWorkingIn ?u}}
```

From these questions, we note the following important points.

- There is no comprehensive analysis on the performance of the relevant questions. Empirically, we can say the performance depends on the interrogated database in the form of storing information, the complexity of the query itself, the optimizations introduced by the query engine and other environmental factors.
- SPARQL does not support the semantics of RDF Schema. For example, the query

```
SELECT ?s
WHERE {?s rdf: type pad: Citizen}
```

- where prefix pad regards the namespace of our ontology and Citizen argument the corresponding class, will return only instances of Class Citizen and not instances of the subclass, as shown in Fig. 11.
- In its current version it is not possible to modify an RDF document, i.e. there are no structures of the respective INSERT, UPDATE, or DELETE the SQL. The SPARQL is currently a pure query authoring language for knowledge acquisition.
- The SPARQL specification defines the query results based on the RDF implication system. Nevertheless, in the specification there is a general, parameterized definition matching model graph, which can be extended to any implication system. Alternatively, the specification acknowledges that inquiries may be made to a virtual graph which has no explicit definition. Using this feature with suitable software, it is possible to produce a graph based on the rules of RDFS implication which are simple SPARQL questions.

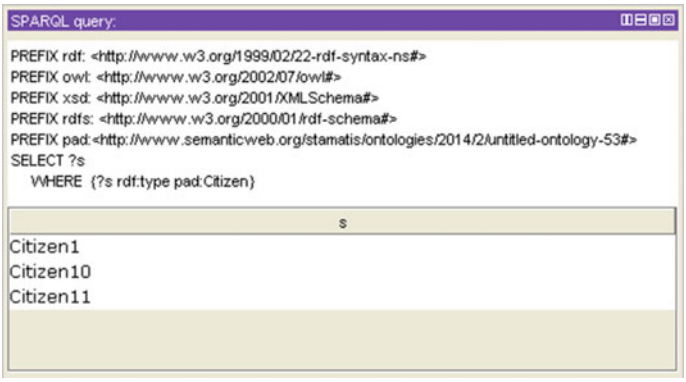


Fig. 11 Members of Citizen class as a response from SPARQL

- As regards the OWL-DL ontology, the OWL-DL axioms do not always lead to a unique graph. Therefore, to answer SPARQL queries to an OWL-DL ontology based on the implication rules of OWL-DL, the implementation of the general customized default template matching securities mentioned is required, suitable for implication based on the rules the OWL-DL. Such an extension of SPARQL is the language SPARQL-DL (which was not developed in this work).

7.2 SPARQL-DL in OWL2 Query Tab of Protégé

SPARQL-DL was designed as an extension of SPARQL, since, as already mentioned, was not supported by the discovery of knowledge from OWL-DL documents (Sirin and Parsia 2007). It appears as a query language capable to retrieve information but also as a separate query submission tool on OWL-DL documents (http://www.w3.org/2001/sw/wiki/SPARQL-DL). SPARQL-DL combines TBox/RBox/ABox queries and templates used by SPARQL to ensure an interoperability form the semantic web.

In our research we installed OWL2 Query Tab as a plugin in Protégé 4.3 and we used it (as well as the DL Query tool and SPARQL tool) for querying the Knowledge base that we have developed. It is a conjunctive query and metaquery engine (for SPARQL-DL and negation as failure), and visualization plug-in. It facilitates creation of queries using SPARQL or intuitive graph-based syntax, and evaluates them using any OWL API-compliant reasoner.

It also provides the possibility of submitting additional atom queries related to the hierarchy of classes as: *directSubClassOf*, *strictSubClassOf*, *directSubPropertyOf*, *strictSubPropertyOf*, *directType*. These questions do not represent ontology data but are implemented through the existing questions of SPARQL.

7.3 DL Query Tool of Protégé

The DL Query tab provides a powerful and easy-to-use feature for searching a classified ontology. It comes with the standard distribution of Protégé Desktop (versions 4.x and above), both as a tab and also as a view widget that can be positioned into any other tab. The query language (class expression) supported by the plugin is based on the Manchester OWL syntax, a user-friendly syntax for OWL DL that is fundamentally based on collecting all information about a particular class, property, or individual into a single construct, called a frame (http://protegewiki.stanford.edu/wiki/DLQueryTab).

With this tool, we are able to submit questions which are based on inference of rdfs shapes. As seen above, the questions of SPARQL retrieve information from the RDF statements in the relevant documents without using inferences. This weakness is covered with the DL Query of the Protégé. The search results correlate triples on the basis of logical conclusions generated by the chain (transitional, reverse, etc.) properties and relations between classes and subclasses. For example, it supports the logic that an instance of a class P is at the same time snapshot of superclass Q. With mathematical symbols would you describe this logic as follows:

$$(w \in P) \cap (P \subset Q) \rightarrow w \in Q$$

In Fig. 12 the reasoners show snapshots of Citizen Class following the relevant question. We note here that the corresponding SPARQL query had no results found.

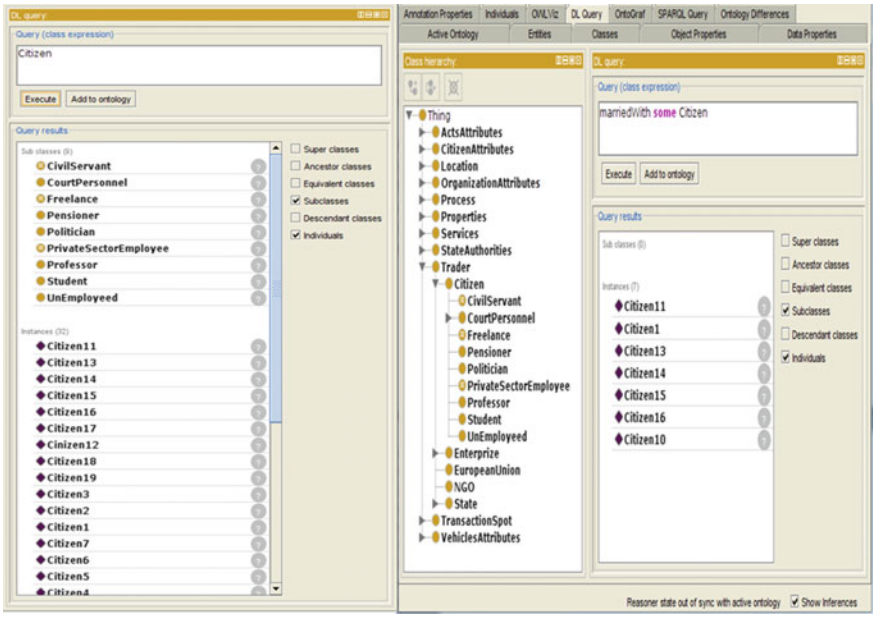


Fig. 12 The DL query results in Protégé

This tool allows the exporting of the results as general axioms in the form of classes. For example the question "*marriedWith some Citizen*" we can define the results in the form of general axiom as shown Fig. 12.

8 Evaluation of Ontology

8.1 Categorization of the Ontology

Generally, an ontology includes a vocabulary of terms and some form of specifications for their importance. Concerning the degree of formality of the representation of an ontology can be:

- Informal, expressed in a natural language.
- Semi-informal, formulated in a limited and structured subset of a natural language.
- Semi-formal, formulated in an artificial and strictly certain language.
- Rigorously formal: definitions of terms with semantics, theorems and proofs of properties such as soundness and completeness.

The ontology developed in this work, is an OWL ontology, i.e. formulated in a certain artificial language. In Protégé environment we have defined a set of rules, axioms and restrictions on classes and properties according to which any stored snapshots are linked. The concepts and properties that we defined follow the semantics based on the actual situation so as listed in the official legal sources. So we could classify our ontology in semi-formal ontologies with several elements from strictly formal ontologies.

A different categorization of ontologies is as follows:

- Knowledge representation ontologies: entities provide representation without specifying what this represent.
- General or common ontologies: aim to capture general knowledge around the world, providing basic concepts such as time, space, events etc.
- Top-level ontologies: provide general concepts under which all terms are associated to existing ontologies.
- Metadata ontologies: provide a vocabulary to describe the information content, which is available electronically.
- Domain ontologies: represent knowledge about a particular field, e.g. medical, administration etc.

Our ontology is at the intersection of general ontologies and of domain of definition ontologies. This is because, in our ontology we have defined various concepts of our everyday life, such as the concept of citizenship, the concept of the state, etc. That is the nature of general ontologies. It also has basic characteristics of domain of definition ontologies, since our main goal was the description of the

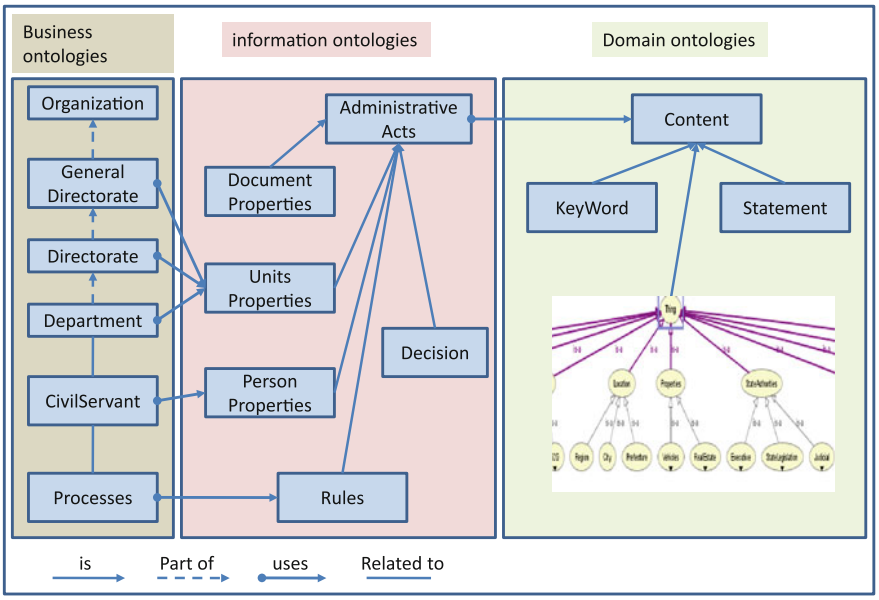


Fig. 13 Categorization of our ontology

scope of public administration and public open data. Finally, our ontology, combines features of business ontologies, information ontologies, as shown below in Fig. 13.

8.2 Basic Principles of Design

When designing the ontology of authentic instruments and public administration, we have considered basic principles (Uschold and Gruninger 1996) of ontologies design, among which we mention the following:

- Clarity and objectivity. The ontology is intended to describe a particular subject field. This field should be attributed clearly and if possible objectively. Our ontology was developed based on existing legislation for the separation of powers, the Greek legislation which defines the criteria of open data and concepts set out in the provisions of administrative law.
- Consequency: the concepts and axioms declared do not contradict or conflict with the axioms or conclusions deduced. This principle was tested extensively using reasoners available in Protégé and analyzed in detail in the preceding paragraph.
- Scalability—reuse of concepts: the design of the ontology should provide scalability and expertise at all levels, i.e. the classes, properties and instances.

This point is important, given that the needs required to cover the ontology may change compared to the original design and that it may be necessary to add a new concept or relationship between concepts. For example, in our ontology in order to describe the concept of "*responsible position Head of Department in the Ministry*" instead of considering a class with all the Heads of Department in ministries (e.g. *HeadOfDtpMinistry*) as individual, we chose to define the following:

- class *Ministry* with individual all ministries
- property *hasPositionInHierarchy*, as a predicate for specific individual having the type *Citizen*
- The individual *HeadOfDepartment*, *type:Bureaucracy*.
- With these definitions we achieve two objectives: (a) achievement of the wanted description and (b) the possibility of reusing of the concepts to describe others.

- Minimal Ontological commitment: The axioms declared should be as generalized as possible. The possibility of extending or application to other fields of interest, shall be given without a clear reference to the specific area covered by the ontology. For example, the statement of the rule that "*whoever works in a department -> also works at the corresponding direction*" was implemented by the rule:
  isWorking (*? x,? y*), *isUnit* (*? y,? z*) -> *isWorking* (*? x,? z*),
  so that it can also be applied in the case where we want to describe that "*anyone who works in an direction -> work and the corresponding general direction*"
- Minimize coding: because the general idea is to reuse the ontology and systems with different approaches, the coding should be as simple as possible and not to affect the information we want to model.
- Semantic analysis of conditions. For example, the concept of citizen is often associated to the property of professional status. Ie we consider a civil servant or a retired or a student etc. is a citizen of a country. So we considered that kinds of professions are subclasses of the Citizen.
- Analysis is aiming to the reuse of concepts.

8.3 Methodology of the Ontology Development

In the bibliography several models of ontologies' development are recorded, depending on the scope and needs to be covered. Among these models we mention the model of Uschold-King (Uschold and King 1995) and the model of Gruninger-Fox (Gruninger and Fox 1995). The development of the ontology in the context of this work follows the basic steps of the model of Uschold-King enriched with elements of the methodology of Gruninger-Fox (Gruninger and Fox 1995). In more detail, the steps we followed in the development of the ontology were:

1. Identification of the purpose for which the ontology was built in conjunction with the capturing of motivating scenarios. The purpose usually arises from the need to provide solutions to specific problems that either have not been resolved or the solutions found do not correspond to reality. In our example, the sector of public administration with the associated public open data is a field that involves many different aspects and concerns such a huge number of concepts and relationships. As mentioned above, the semantic web is expected to provide solutions in this area.
2. Ontology Building
    a. Standardization of informal competency questions (formulation of competency questions) is considered to be important. These questions are based on motivating scenarios and form a sort of requirements to be met by the ontology. These questions must be answered with the terminology developed in the ontology.
    b. Conception of ontology. At this stage we make the identification of the most important concepts and relationships between them and then we make their registration.
    c. Definition of axioms. The axioms define any restrictions of the concepts combined with the atypical sufficiency questions.
    d. Coding. It is done in the Protégé environment with the help of Description logic.
    e. Integration of existing ontologies. We examined already published ontologies in the field of public administration with a view to incorporating any concepts or relationships. No models that align with our model were found.
3. Assessment of the ontology. Besides texts of the documentation of the ontology, the evaluation record results from the use of reasoner featured by Protégé. Their use enables us to ensure any issues of inconsistency restrictions and general axioms.
4. Maintenance of the ontology. In the final stage, which may be the first of any revision of the ontology, we considered any omissions or the need to adapt to new requirements.

9 Conclusions

In this paper we have presented issues concerning the application of Semantic Web technologies in the area of e-government and open data. This area is characterized by the huge number of primary data which are open public data, by the failure to observe common vocabulary for the concepts and procedures used throughout the public sector, and by the insufficient set of information retrieval tools of the public sector. The technologies that have been introduced aim to support the semantic approach in the seeking of information in the vast area of open public data. They are

the infrastructure for the development of new intelligent applications that will use both the public and the same public administration for the management of open public data. From the study which was preceded we can point the following conclusions.

- The Protégé's assistance in the development of the ontology was important. This program which constitutes free open source software, provides the ability to manage ontologies both local on the computer of each user and online. In its last versions it is a very user-friendly tool developing only OWL ontologies, with query reporting possibilities with two main tools: the DL Query oriented exploitation of semantic expressiveness of OWL and SPARQL oriented simple data in RDF format.
- The Protégé comes with different versions with the main 3.4.8 and 4.x. Key differences are recorded to support OWL 2 versions 4.x, non-supported version RDF documents and OWL-Full in newer versions (http://protegewiki.stanford.edu/wiki/Protege4Migration#Side_by_Side_Comparison). However, the existence of different technologies and versions of OWL (OWL-Full, DL OWL 2) creates compatibility problems and confusion between different versions of Protégé and other related tools.
- Significant is the assistance of reasoners for debugging and correcting logical errors in addition to their basic function is to support the logical inference basis of OWL expressive possibilities.
- The methodology followed for the development of our ontology, is a combination of two basic methodologies that are used in numerous ontologies.
- Since the Protégé does not support the Greek language, the concepts we have set have been translated into English. The issue of supporting the concepts in Greek language is open and its resolution in the future will contribute to the further development of semantic web applications.
- There is a clear need to develop online queries applications using SPARQL-DL to support the submission of intelligent queries that exploit the increased semantics offered by OWL ontologies.

References

Goudos, S.K., Peristeras, V., Loutas, N., Tarabanis, K.: A public administration domain ontology for semantic discovery of eGovernment services. ICDIM (2007)

Gruninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal (1995)

Heath, T., Bizer, C.: Linked data: evolving the web into a global data space. Synth. Lect. Semant. Web: Theory Technol. **1**(1), 1–136 (2011). Morgan & Claypool. Electronic version in http://linkeddatabook.com/editions/1.0/

Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: The Semantic Web—ISWC 2010, Lecture Notes in Computer Science, vol. 6496, pp. 402–417 (2010)

O'Connor, M.: The semantic web rule language, Stanford University, in http://protege.stanford.edu/conference/2009/slides/SWRL2009ProtegeConference.pdf (2009)
Parundekar, R., Knoblock, C.A., Ambite, J.L.: Linking and building ontologies of linked data. In: The Semantic Web—ISWC 2010, Lecture Notes in Computer Science, vol. 6496, pp. 598–614 (2010)
Sirin, E., Parsia, B.: SPARQL-DL: SPARQL Query for OWL-DL, 3rd OWL Experiences and Directions Workshop (OWLED-2007) (2007)
Theocharis, S., Tsihrintzis, G.: Ontology development to support the Open Public Data—The Greek case. IISA (2014)
Uschold, M., Gruninger, M.: Ontologies: principles, methods and applications. Knowl. Eng. Rev. **11**(2), 93–136 (1996)
Uschold, M., King, M.: Towards a methodology for building ontologies. In: IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal (1995)
www.w3c.org/owl



http://www.springer.com/978-3-662-49177-5

Intelligent Computing Systems
Emerging Application Areas
Tsihrintzis, G.A.; Virvou, M.; Jain, L.C. (Eds.)
2016, X, 368 p. 144 illus., 124 illus. in color., Hardcover
ISBN: 978-3-662-49177-5