algorithms

MDPI

Article

Ontology Based Governance for Employee Services

Eleftherios Tzagkarakis [1], Haridimos Kondylakis [1,2,*], George Vardakis [1] and Nikolaos Papadakis [1]

[1] Department of Electrical and Computer Engineering, Hellenic Mediterranean University (HMU), Estavroménos, 71410 Iraklio, Greece; eltzaster@gmail.com (E.T.); vardakisyageorge@hmu.gr (G.V.); npapadak@hmu.gr (N.P.)
[2] Foundation of Research & Technology-Hellas (FORTH), Vassilika Vouton, P.O. Box 1385 Heraklion, 70013 Crete, Greece
* Correspondence: kondylak@ics.forth.gr

Abstract: Advances in computers and communications have significantly changed almost every aspect of our daily activity. In this maze of change, governments around the world cannot remain indifferent. Public administration is evolving and taking on a new form through e-government. A large number of organizations have set up websites, establishing an online interface with the citizens and businesses with which it interacts. However, most organizations, especially the decentralized agencies of the ministries and local authorities, do not offer their information electronically despite the fact that they provide many information services that are not integrated with other e-government services. Besides, these services are mainly focused on serving citizens and businesses and less on providing services to employees. In this paper, we describe the process of developing an ontology to support the administrative procedures of decentralized government organizations. Finally, we describe the development of an e-government portal that provides employees services that are processed online, using the above ontology for modeling and data management.

Keywords: administration ontology; e-government; OWL; public administration

check for updates

Citation: Tzagkarakis, E.; Kondylakis, H.; Vardakis, G.; Papadakis, N. Ontology Based Governance for Employee Services. *Algorithms* **2021**, *14*, 104. https://doi.org/10.3390/a14040104

Academic Editor: Frank Werner

Received: 17 February 2021
Accepted: 22 March 2021
Published: 25 March 2021

1. Introduction

The progress made in the field of computers and among them in the field of networks and the internet has significantly affected all areas of our daily lives. Our daily activities have the potential to be simplified and performed more efficiently. This progress has affected not only the operations of the private sector but also the operations of the public sector by introducing the concept of e-government. In this context, governments are trying to formulate a central digital policy, by delineating the axes and determining the directions for its development. Their goal is to increase and improve the services provided electronically, with minimal physical interaction, thus reducing administrative burdens. Special emphasis has been also placed on facilitating access to electronic services through more user-friendly interfaces and one-stop government points.

The establishment of e-government requires the transformation of public administration through the simplification and digitization of administrative procedures. The most effective operational planning presupposes the understanding and modeling of the correct needs of the procedures. This will lead to the necessary redesign of the procedures in order to optimize the provided electronic services.

This transformation is not a simple process. The public sector has a huge range of functions and services offered to citizens and businesses. The structure and operation of the public sector make reforms time-consuming and complex. Reforming usually requires changes in the legislation and a series of circulars that will regulate the individual issues that arise. Besides, the rapid introduction of computers in society in recent decades has created the need for the immediate introduction of computers in public administration. This resulted in the development of systems oriented to the needs of each organization before the central government had time to adequately formulate and implement a central digital policy. Many of the developed systems attempted to digitalize existing processes without having undergone the necessary administrative transformations. Therefore, these systems face difficulties in cooperating, showing a high degree of heterogeneity. Besides, these systems face difficulties in adapting to the newly reformed framework set by the central government.

By careful examination of the way e-government has developed in recent years, we can verify that the key priority is the provision of services to citizens (government to citizen) and businesses (government to business). However, another area of e-government that has not been explored extensively is the provision of electronic services to employees (government to employee). The low priority given to this area is normal, as the administrative burden of handling services to civil servants is generally lower. On the other hand, there are cases of public organizations with a large number of employees and a large geographical spread. In this case, the number of employees makes the administrative burden for their management considerable.

At a theoretical level, the need for online governmental services has been widely recognized. There have been many initiatives in recent years around the world and several projects have been developed to promote e-government. However, as [1] and later [2] point out, there is a gap between theory and practice. Despite the plethora of work and suggestions on the use of ontologies in e-government, the percentage of works that have suggestions for practical applications is low. On a practical level, many organizations have set up websites establishing an electronic interface with the citizens and businesses with which they transact. Most agencies, especially the decentralized ministry agencies and the local government agencies, provide mainly information services (stage 1 services) and printed applications in electronic form (stage 2 services). In these cases, citizen service begins and ends nonelectronically. Few organizations provide services that are started electronically and completed manually (stage 3 services) and even fewer provide services that are completed entirely electronically (stage 4 services). In addition, few agencies have been involved in offering integrated services to their employees.

Knowledge management is of particular importance for the development of a system. Traditional knowledge representation systems have a centralized structure and impose strict rules for data representation. This results in reduced flexibility and soon leads to management problems. Besides, it limits the information we can retrieve and its scalability [3].

Knowledge management in e-government applications must take into account its online nature. A key element is the need for user interaction. This interaction replaces the physical transaction of the citizen with the state and offers integrated services to the citizen. Developments in the field of the World Wide Web promote the interaction of users through the Social Web (Web 2.0). However, the amount and structure of data generated with Web 2.0 are not properly structured to facilitate data interconnection and reasoning [4,5].

The Semantic Web (Web 3.0) adds logic to the web and provides a solution to these problems. The Semantic Web is an extension of the Web that organizes data in a formal, structured form and sets inference rules [3]. Ontologies depict a shared, agreed, and detailed model (or set of concepts) of a domain. Another advantage of using a domain ontology is the definition of semantic models of the data combined with the associated domain knowledge [6]. In general, knowledge management becomes more efficient when using the linked data technology stack. It is important that the data is machine-readable and ensures interconnection with other data sets [7] Formal ontologies help the data to be represented at a higher conceptual level, regardless of the task they are called to perform [8]. The combination of Social and Semantic Web is essential for the effective implementation of e-government.

Remarkable is also the research in the field of ontologies in e-government. The majority of these proposals use a top-down approach and are mainly concerned with defining horizontal ontologies.

Besides, there are few e-government applications based on ontologies. Finally, it is noteworthy that despite the research carried out, there is difficulty in defining backbone ontologies by governments.

In this paper, we present on the one hand the development of an ontology to support administrative work and on the other hand, a good practice for the process of developing an e-government portal that will provide comprehensive services to employees. The portal is based on the proposed ontology, both for the modeling and for data management. We followed a bottom-up approach, starting with the analysis of the requirements of the organizations and continuing with the design and implementation. The decentralized Primary and Secondary Education Agencies under the Ministry of Education of Greece were selected as the scope of application. These organizations have a common administrative framework and are called to serve the needs of more than 200,000 permanent and contract teachers in Greece.

Within this work, we propose a model for the development of e-government, through the development of individual specialized vertical ontologies developed with a bottom-up approach and the interconnection between them. The contribution of this work lies in exploring the development of ontologies for government to employee. Within our work, we highlight the effectiveness of ontologies in developing web applications for e-government. We also focus on the usefulness of vertical ontologies in the provision of integrated electronic services for both modeling and mainly for data management, by presenting a good practice model for e-government.

The rest of the paper is structured as follows. In Section 2 we present relevant work in the field of e-government. In Section 3 we present our methodology. Section 4 describes the Domain of the Ontology and Section 5 presents the Web Portal constructed on and the lifecycle of a user's request. Finally, Section 6 concludes this paper and presents directions for future work.

2. Related Work

In the last few decades, there has been an increase in semantic web ontologies that try to model the services offered by the public sector. In [9] we see a study about knowledge management, which can be used for designing and developing e-government services. They suggest the use of knowledge units which, through a domain map, are related to the transaction service components that will be implemented. In the context of the SmartGov project, they created a general-purpose ontology that aims to provide a conceptual framework at the cognitive level and not a special-purpose ontology.

Another project that recognizes the need for proper knowledge management and deals with the conceptual level is the OntoGov project [10,11]. This project tries to address the problems that arise in the provision of services to public bodies due to the frequent changes that take place in the legislation. For this purpose, they defined a cluster of ontologies for modeling e-government services. The overall goal of the OntoGov project is to develop, test, and validate a semantically enriched platform that will facilitate the consistent restructuring of e-government services.

The ICTE-PAN project dealt with the support and structure of the highest-level functions of the government, such as the planning, implementation, and evaluation of public policies for major and complex social problems. In this work [12] distinguished the need to interconnect heterogeneous systems with different backgrounds, interests, and values. He developed a horizontal ontology which, however, can be combined with vertical ontologies in the management of specialized topics.

The Governance Enterprise Architecture (GEA) [13] is a top-level enterprise architecture that aims at the overall description of a governmental model. They suggest five levels for process and object models, the GEA mega-process model, the GEA interaction model, the GEA public policy formulation object model, the GEA service provision object model, and the latest development of the GEA object model for the overall governance system. According to the proposed model, the interaction between the administration and the citizens is divided into two main parts. The first one, the planning part, consists of the necessary actions needed to provide citizens with the necessary information to identify and use the available services. The second consists of the necessary actions to provide the citizen with the product of the service. Their research continued with the implementation of the model using OWL [14]. In this context, they developed an application that attempts to provide the user with access to related functions, based on his profile. They also explored the conceptual mappings between GEA entities and Web Service Modeling Ontology (WSMO) service model components [15].

GEA was the basis for other works in the development of ontologies, particularly in the area of service detection by citizens. The OntoAL ontology [16] attempts to adequately describe the state structure of Albania, making the necessary modifications to adapt to local conditions. This ontology aims to translate public services into daily activities so that the citizen can easily find services even if they have not been modeled.

The authors in [17] used a model proposed in GEA and constructed a semantic model to support the identification of the services by the citizens. They proposed a model-driven architecture methodology and they built a framework for the needs of a municipality. Its purpose was to assist users in the search for services and to provide them with relevant access to it, making them independent of domain experts.

There are also many more top-level ontologies for e-government. Indicatively, we mention some of them. The authors in [18] proposed an e-government framework in OWL that uses ontologies for the State of Kuwait. The ontologies are designed for assisting interoperability and interconnecting information from different government organizations. They integrated information from the domains of health care, education, and civil information, taking into account the common information from all these domains. IndiGov-O is another top-level ontology that conceptualizes the structure of government in India. The authors in [19] introduce a four-level hierarchy to represent the ministries and their departments that will form the basis for future extension of e-government.

Apart from top-level ontologies, there is interesting research for the development of vertical ontologies in specific sectors of public administration. In the field of legal ontologies, [20] introduced an e-government ontology model for the real-estate transaction domain, in Spain. This model is a part of the e-Government Ontology (EGO) model that was developed within the Reimdoc Project. The aim of the project was the modeling of legal documents and information to enable their retrieval during the processing of transactions with citizens. This issue has also been investigated in other works [21,22]. The eGRRC framework that was proposed by [23] can be used for modeling legislation but mainly focuses on regulatory requirements compliance and their interrelationships.

Another area that has attracted the interest of researchers is financial e-government ontologies. In [24], the authors presented an ontology for the budgetary and financial system of Santa Fe province in Argentina. They combined ontology development methodologies and software engineering techniques to highlight the advantages of ontology-based applications. Another financial ontology, focusing on public procurement, is the Public Procurement Ontology (PPROC) [25]. The PPROC ontology covers all the stages of procurement processes and contracts. The Zaragoza's City Council and the Provincial Government of Huesca have adopted it. PPROC ontology complies with transparency laws providing open data for public procurement.

Other researchers have also pointed out the inadequacy of ontologies in open government and open data. In [26], the authors developed a transparency ontology with OWL and Protégé. They started their research driven by the deduction that there is difficulty in selecting the data that will be used to create an effective ontology. Therefore, they focused their research on which data is more understandable to citizens, in order to select an optimal set for the ontology. An innovative work was developed at Rensselaer Polytechnic Institute in the field of linked open government data [27]. The result of this work was the Tetherless World Constellation (TWC) of Linked Open Government Data (LOGD) portal, a Semantic Web-Based platform for sharing open data. The approach for the LOGD portal was adopted in the UK and USA for their open data portals. In the UK (https://data.gov.uk) (accessed on 24 March 2021) there was a top-down implementation, while in the USA (https://www.data.gov) (accessed on 24 March 2021) they preferred a bottom-up method.

In [22], another vertical ontology is proposed that aims to support integrated services of a public organization. The ontology consists of the structure of public administration in Greece and the documents that are used. The documents are divided into the documents that are produced by the administrative procedures and the documents that concern the legal framework that is applied. They presented the application of the relevant ontology in the administration procedures of the local authorities of Greece.

Ontologies, beyond their effectiveness in imaging and modeling systems for developing applications that implement services, may be particularly useful for the interconnection of heterogeneous systems. There is a lot of relevant work on this subject. Indicatively we mention some of them. The authors in [28] presented a semantic-based architecture of a one-stop government portal, which on the one hand supports the user in searching for services on interconnected systems and on the other hand ensures the processing of the requested service. EG-BOnt was another approach by [29] for modeling and collaboration on the government to business field. The authors in [30] proposed an e-government interoperability framework in Uganda. They designed a National Enterprise Architecture that is based on a set of related ontologies.

The contribution of the above works is particularly useful and necessary for the mapping of how ontologies can be utilized in e-government. These papers use more a top-down approach and focus on top-level and horizontal ontologies. Most of them try to give general directions for the development and use of ontologies in public administration. Despite the relevant research, governments around the world face difficulties in defining a central governmental ontology that will support the development of electronic services. Besides, there are not many vertical ontologies, especially in the field of government to employee.

The experience gained from the above work is the basis for the next step in this area. Our proposal is based on the development of vertical ontologies that will result from a bottom-up approach. Each organization or group of organizations with a common working framework must design the ontology that will support its administrative procedures. Beyond that, the emphasis should be placed on the interconnection of these ontologies to achieve the desired degree of interoperability.

The heterogeneity that results from this process should not be considered as a deterrent. Given the general difficulty of defining and applying common standards, heterogeneity is part of the system and is already a phenomenon that is being addressed. Heterogeneity is a more general problem that does not occur only in our case. Additionally, the modeling offered by the use of ontologies helps us to overcome the issues of heterogeneity more easily. Several papers suggest solutions to this issue through the use of ontologies [28–30].

For our application, we used ontologies. Ontologies offer efficient modeling of structures, data, and services. This, in addition to the advantages it offers us for design and implementation, facilitates the interconnection of heterogeneous systems. Moreover, the reasoning capability enables us to infer information more easily from our data.

The solution we propose can also be considered in conjunction with other research papers that explore the potential for exporting data from relational databases into ontologies [6,31–34]. In this way, it is possible to utilize the huge volume of data that already exists in relational databases, to facilitate the transition of information systems using relational databases to semantic ontology systems.

The purpose of our work is to create an ontology to support administrative work and an ontology-based e-government portal that offers integrated services of stages 1–4 to employees. In this context, the ontology manages the modeling and management of knowledge, which is used in the services of all four stages. The Web Portal, for its part, undertakes the implementation of the services, offering the necessary interfaces to the clients (employees) for the submission of requests to the administration and the receipt of the results. It also provides the necessary interfaces for the administration to access the SPARQL endpoint for data management and request processing.

For the construction of the ontology, we chose the OWL language, because of its stronger semantics and logic relation expressiveness. For the implementation of the ontology we used Protégé 5.5.0 [35] and we verified the ontology with the Pellet reasoner [36]. In addition, we checked the ontology for inconsistencies and pitfalls with the OntOlogy Pitfall Scanner (OOPS) [37]. The portal was built with open source tools. The web interface was powered by WordPress and we used PHP custom templates that were handling the user requests and the SPARQL queries. We also set up a SPARQL Endpoint with Apache Jena for data management.

3. Methodology

There are several approaches for designing and implementing e-government ontologies. In general, the methodologies are divided into two categories: top-down and bottom-up [38,39]. In the first category, top-down methodologies attempt to cover the entire range of a domain by describing the concepts and relationships between them. These methodologies first model high-level concepts and subsequently they fine-grain them and go deeper into lower levels. According to [39], this approach has the advantage that the ontologies that are created can easily be reused and can be used as a basis for creating new ontologies. These ontologies also contain a semantically rich vocabulary of terms and relationships and offer better control of the level of detail. The top-down development of an ontology has the disadvantage that it requires a lot of time and effort. Designing an ontology with this methodology requires very good knowledge of the domain to produce adequate modeling. Moreover, the ontologies that are built with this methodology are more general and face difficulties in managing changes.

On the other hand, in bottom-up methodologies, the modeling of concepts starts from the lower levels. In this approach, the concepts are easier to identify and the ontology is built faster. This methodology more easily identifies the biases and inconsistencies caused by humans. Moreover, the ontologies are modified more easily and respond more quickly to changes that take place [39]. Their disadvantages [39] include the fact that they have a high degree of detail, which creates problems in identifying common correlations and increases the risk of inconsistencies [40]. This approach does not use a common model and shape to represent the concepts. In addition, each ontology uses a different model for its development [29].

To develop our ontology we chose to use a bottom-up approach. Our ontology aims to serve the needs of a specific group of organizations. Despite the research that has taken place in recent years, the state has not adopted top-level ontologies that could stand as the backbone for our effort. Our goal was to create a vertical ontology that is flexible and can be easily extended to adapt to frequent changes.

Initially, we performed a detailed mapping of the structure and function of the organizations. In order to record the procedures that are performed, interviews were conducted with domain experts (i.e., heads and employees of administrative departments). During the design, we started from the specific tasks we want to implement, taking into account the possibilities of extending the ontology. As [41] proposed, we developed our ontology incrementally. In the first steps, we integrated certain concepts and relationships and continued with defining new ones, or creating instances as needed.

More specifically, we followed the following steps:

1. **Mapping status.** In this step we depicted the structure of the decentralized agencies and identified the processes that take place in these organizations.
2. **Selecting procedures** for implementation. We ranked the processes of the previous step using as criteria the number of individuals involved and the repetition period of each process, to select those that cause the greatest administrative burden. For each of the selected procedures, we followed steps 3 to 8.

3. **Analyzing procedures** by identifying the events (triggers) that initiate the procedures, the individuals and organizations involved in the procedures and the additional data that need to be retrieved from the organizations' files. We also analyzed the data processing that takes place and the outputs of the process.
4. **Ontology Design.** We modeled the entities and separated them into "Organizations", "People", "Documents", and "Data" participating in the processes.
5. **Ontology Building.** We implemented the ontology with the Protégé tool.
6. **Ontology Population.** To verify the functionality of the ontology we created a set of individuals.
7. **Procedure integration in the Portal.** We developed the user interfaces for each procedure and we added the necessary code to communicate with the SPARQL Server.
8. **Testing the Web Portal.** During the testing of the functions in the Web Portal, we checked the correctness of the design and implementation of the Ontology and proceeded to corrections where necessary, returning to Step four.
9. **Ontology Evaluation.** Evaluation of the ontology and checking for inconsistencies and pitfalls.

Focusing on the procedures implemented we have to note that are mainly divided into "procedures involving human intervention" (e.g., granting leave) and "fully automated procedures" (e.g., issuance of a certificate).

The triggers that initiate the procedures are mainly the will of the employees that are expressed by submitting an application. The application is an incoming document that contains the details of the applicant, identifies the requested service, and the necessary parameters required to process the application. Some other procedures are initiated ex officio by an organization as they fall under its statutory obligations, such as hiring or firing contract teachers.

The data needed to complete a process consist of the data that are kept in the agency's files for employees (e.g., for the teachers, the schools they have served).

Process outputs are documents that validate an act or certify a situation. To create these documents, we have to collect the necessary data concerning the employee. After this, we have to connect them with the details of the organization that issues the document, the supervisor who certifies them, and the employee who compiles the document. The administration's decision is also recorded in case it is required (e.g., the approval or rejection of an application of leave).

4. The ADM Ontology

Following the previous analysis, we proceeded to the design of the ontology. The ADM Ontology contains 71 classes, 41 object properties and 53 data properties.

4.1. Classes

The entities that we represent are classified into the following classes:

- Organization
- People
- Document
- Data

4.1.1. Organization

The Organization class (Figure 1) is used to represent Organizations and their structure. It is divided into three main subclasses: Administration, School, and Other Organization, which are disjoint. The Administration subclass contains the Administrative Organizations and their hierarchical structure is reflected with the relevant object properties (supervises, isSupervisedBy) that show which organization supervises which. The Ministry supervises the Regional Directorates and they supervise the Directorates. Each organization is supervised by only one organization of the higher level, while it can supervise several lower level organizations.

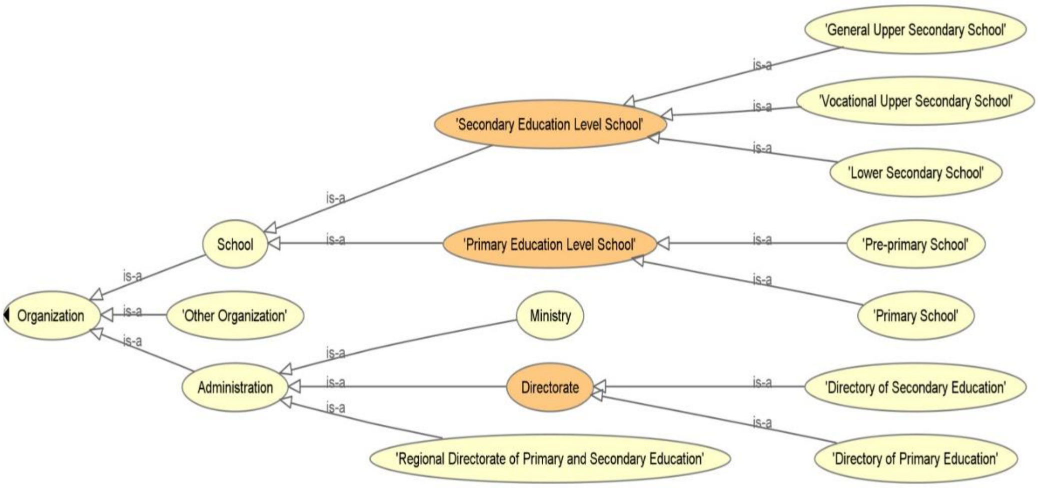

'General Upper Secondary School'
'Vocational Upper Secondary School'
'Lower Secondary School'
'Secondary Education Level School'
School
'Primary Education Level School'
'Pre-primary School'
'Primary School'
Organization
'Other Organization'
Ministry
Administration
Directorate
'Directory of Secondary Education'
'Directory of Primary Education'
'Regional Directorate of Primary and Secondary Education'
is-a

Figure 1. The Organization class hierarchy as shown in Protégé.

The School class is used to classify school units. All subclasses of the class School are disjoint to each other. Depending on the type of school unit (primary or secondary), it is supervised by a Directorate of the respective level. Each school unit is supervised by only one Directorate. Finally, the class Other Organization is used to represent organizations that do not belong to the Body of the Ministry of Education, but it is necessary to record them as they have issued acts in the past that affect the official status of our employees.

An organization is run by a director (hasHead property) and has at its disposal the employees who serve in it (hasEmployee property). An organization also receives incoming documents from people and organizations (hasReceived property) and issues documents (hasIssued property) that communicates to people and organizations. Civil Servants also offer service to an organization (hasReceivedService), which is recorded for its personal records.

The Organization class has 15 subclasses, uses 12 object properties 7 data properties and inheritance depth 3.

4.1.2. People

The class people (Figure 2) is divided into 2 main subclasses, Citizen and Civil Servant. The Civil Servant class contains the subclasses Director, Administrative Employee, and Teacher. The class Administrative Employee refers to civil servants who belong to Administrative Organizations, while the class Teacher refers to civil servants who belong to a school. The above two classes are disjoint, while the members of the class Director can be either Administrative Employees or Teachers. The class Director contains the civil servants, that are head of an organization and have the authority to sign the issued documents. To state that a civil servant is director of an organization we use the "isHeadOf" object property.

An instance of the class People can receive Outgoing Documents from an Organization and can sign a document. The sign procedure is depicted with the "hasSigned" object property. The action of signing a document is general. It can depict many physical actions of a person depending on his role. A citizen or a civil servant signs an application that submits to organization. A civil servant signs an outgoing document that he processes and a director signs an outgoing document issued by his organization.

Since our ontology is oriented towards government to employee support, civil servants play different roles depending on the occasion. When a civil servant acts as a client of services can request a leave or a certification. When administration issues a decision or a certificate has to take into account information stored in the personal records of the civil servant. With the relevant object properties we correlate a civil servant with an Organization (isEmployeeOf property), with changes in his official status (isEffectedBy property), with his working experience in other organizations (hasPreviousService property), and with leaves granted so far (hasRequestedLeave property).

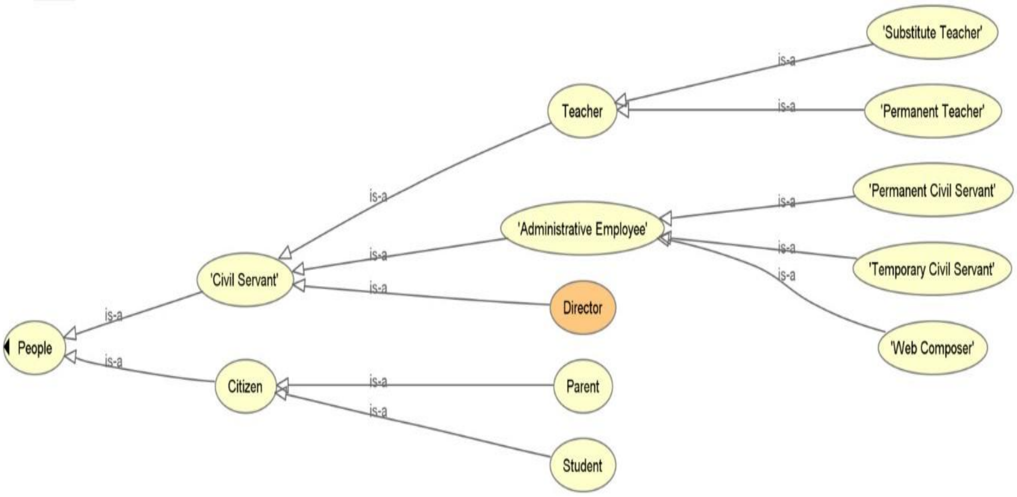


Figure 2. The People class hierarchy.

When a civil servant acts as a representative of the administration composes outgoing documents (hasComposed property), represents an organization (isEmployeeOf property) or signs an issued document (hasSigned property) if he is a director.

The class Citizen is added for future extension of the ontology and is divided into Parent and Student. Since the vast majority of students attending primary and secondary schools are minors, until adulthood their transactions with public organizations are carried out through their parents.

The People class has 12 subclasses, uses 16 object properties 7 data properties and inheritance depth 3. The subclass Civil Servant has 3 more data properties.

4.1.3. Document

Documents, as shown in Figure 3, are divided into two major categories which are represented by the respective classes: Incoming Documents, and Outgoing Documents. There are many types of Incoming Documents but, at this stage, we only use one type, the applications. The Outgoing Documents we use are certificates and decisions. A document is signed by a person (isSignedBy property). An incoming document is submitted to an organization (isForwardedTo property), while an outgoing document is communicated to some Organizations or People (isForwardedTo property).

For each type of application, there is the relevant type of outgoing document. Each incoming document is correlated with the outgoing document that processes it, with the "isProcessedBy" property. When an application is submitted an instance of the application is created to the relevant class, according to the type of the application. In cases where a decision of the administration is required regarding the approval of the application or not, an additional instance is created for the object of the application. For example, an application for leave creates an instance in the leave class. This instance is connected to the application with the "isRequestingFor" property and to the decision with the "isRefferedIn" property.

When an application is processed, an outgoing document is issued by the organization (isIssuedBy property). The document is composed by a civil servant (isComposedBy property), is signed by the director (isSignedBy property) and forwarded to the recipient (person or organization) (isForwardedTo property).

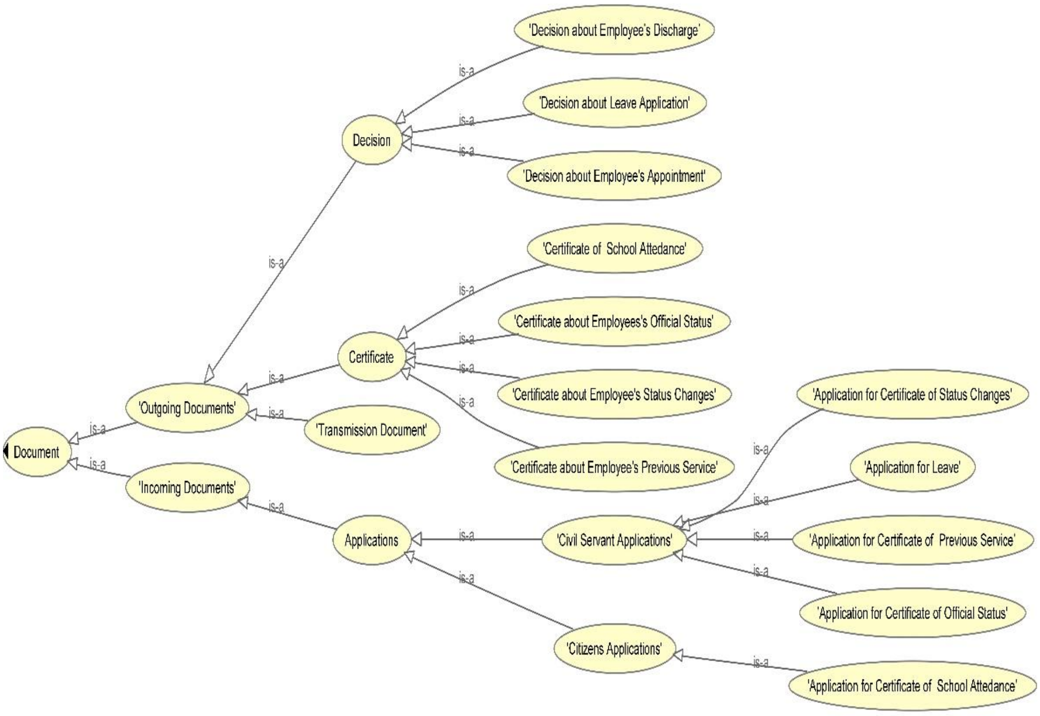

'Decision about Employee's Discharge'
'Decision about Leave Application'
Decision
'Decision about Employee's Appointment'
'Certificate of School Attedance'
'Certificate about Employees's Official Status'
Certificate
'Certificate about Employee's Status Changes'
'Certificate about Employee's Previous Service'
'Outgoing Documents'
'Transmission Document'
Document
'Incoming Documents'
Applications
'Civil Servant Applications'
'Citizens Applications'
'Application for Certificate of Status Changes'
'Application for Leave'
'Application for Certificate of Previous Service'
'Application for Certificate of Official Status'
'Application for Certificate of School Attedance'
is-a

Figure 3. The Document class hierarchy.

We have also created the subclass of the transmission document for future use, as it is a basic type of document used in public administration.

The Document class has 20 subclasses, uses 16 object properties 3 data properties, and inheritance depth 4. The subclass Application has 1 more data property.

4.1.4. Data

In the Data class (Figure 4), we model the data we manage that are necessary for the processing and the production of outgoing documents. The data of each class is used to handle more than one process and this modeling can be used in future system extensions. The data is connected to the outgoing document with the "isRefferedIn" property.

The "Changes in Civil Servant's status" class contains the data that is kept in the personal records of the employees. These data are useful for the issuing of almost all documents for Civil Servants. The changes are linked to the civil servant with the "hasEffectOn" property.

The "Previous Service" class contains information related to the services of the staff before and after the appointment. Each service is connected to a civil servant with the "wasOfferedBy" property and to an organization with the "wasProvidedIn" property.

Finally, the "Leave" class classifies the types of leaves. For each type of leave there is a separate subclass. The leave instance connects to its application with the "isRequestedWith" property and with the civil servant with the "isRequestedFrom" property.

The Data class has 20 subclasses, uses 12 object properties, and inheritance depth 3. The subclasses of class Data have 18 data properties.

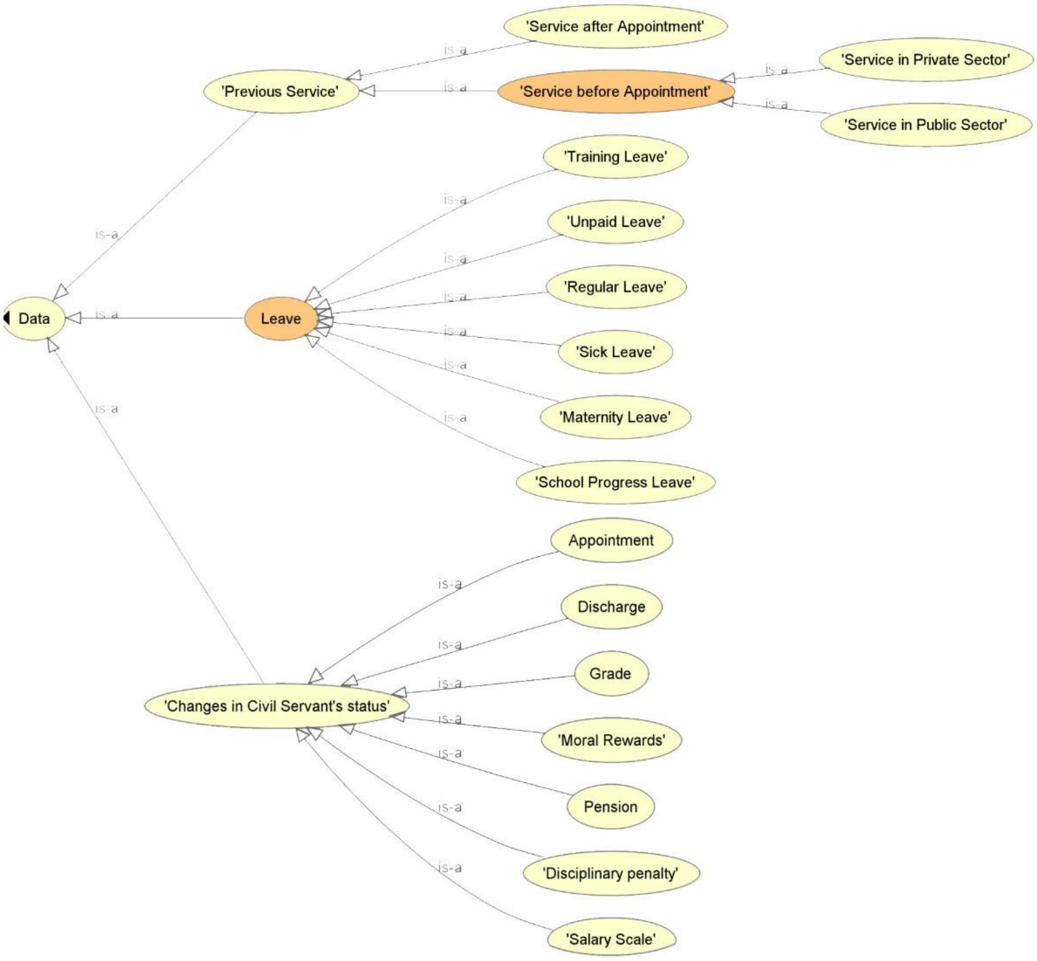


Figure 4. The Data class hierarchy.

4.2. Object Properties

The object properties depict the hierarchy of organizations, the relationship between people and documents, people with data, people with organizations, organizations with data, organizations with documents, and documents with data. In Table 1 these relationships are presented.

Table 1. Object Properties.

		Range			
		People	**Organization**	**Document**	**Data**
Domain	People		isHeadOf isEmployeeOf	hasSigned hasComposed hasReceived	hasRequestedLeave hasPreviousService isEffectedBy
	Organization	hasHead hasEmployee	supervises isSupervisedBy	hasIssued hasReceived	hasReceivedService
	Document	isForwardedTo isComposedBy isSignedBy	isForwardedTo isIssuedBy	hasProccessed isProccessedBy	hasReffered isRequestingFor
	Data	hasEffectOn isRequestedFrom wasOfferedBy	wasProvidedIn	isRefferedIn isRequestedWith	

4.3. Data Properties

In Table 2 we can see the data properties of the individuals that belong to the classes mentioned in Section 4.1. The Application class is a subclass of the Document class, so its individuals inherit the parent class as well. Respectively, the Civil Servant class is a subclass of the People class and its instances inherit, respectively, from the parent class. The Changes class is designed to include all types of status changes. The Data Properties "hasChangeDate", "hasDecisionAuthority", "hasDecisionDate" and "hasDecisionRefNum" are common to all types of changes. The rest properties depend on the type of the change.

Table 2. Data Properties.

Class	Data Properties
Organization	hasOrgId, hasTitle, hasAddress, hasTel, hasFax, hasEmail, hasWebSite
Document	hasDate, hasRefNum, hasTheme
Application	hasAppStatus
People	hasSurname, hasFirstName, hasFatherName, hasMotherName, hasIdentityCardNum, hasTaxIDNum, hasBirthDate
Civil Servant	hasRegistryNumber, hasEducationLevel, hasSpeciality
Previous Service	hasServiceTitle, hasServiceStartDate, hasServiceEndDate, hasServiceTotalTime, hasServiceHeadPosition
Leave	hasLeaveStartDate, hasLeaveEndDate, hasLeaveNumOfDays, hasLeaveStatus
Changes	hasChangeDate, hasDecisionRefNum, hasDecisionDate, hasDecisionAuthority
Discharge, Appointment, Pension	hasGovGazzeteNum
Grade	hasPenaltyType
Disciplinary penalty	hasPenaltyType
Moral Rewards	hasRewardType
Salary Scale	hasSalaryScale

4.4. Validation and Evaluation of the Ontology

For the validation of the ontology, we first used the Pellet reasoner. Pellet is a complete OWL-DL reasoner. It is open source and it accessible through various interfaces. In our case, we used the Pellet plugin of Protégé. As its developers [36] state "It offers a panoply of features including conjunctive query answering, rule support, e-connection reasoning, and axiom pinpointing, among others". Using Pellet, we checked the ontology for inconsistencies and errors that may have occurred during the design and implementation.

We also checked the ontology for inconsistencies and pitfalls with OOPS (OntOlogy Pitfall Scanner) [37]. OOPS is a tool for detecting pitfalls in ontologies. It checks an ontology and predicts potential problems that may arise, based on a list of bad practices "pitfalls". OOPS tries to identify features that often represent a problem or that could lead to ontology errors. For our case OOPS detected minor problems which were corrected and as a result no problems are currently reported by the tool.

Finally, we evaluated the ontology through its real-world usage in the web portal, and we found it adequate and complete, completely covering the cases for which it has been developed. A survey was conducted in which 42 people used the Web Portal and then completed a questionnaire in which they recorded their experience of using it. The participants in the research were civil servants, administrative employees, and teachers, who serve in decentralized organizations of the Ministry of Education. As the ontology is oriented towards the development of functions in the field of government to employee, the participants had the advantage that they could participate in a dual role. More specifically, they were invited to use the Web Portal both as users who wish to submit their requests, and also as representatives of the administration that needs to process the requests that have been submitted to the organization. For this purpose, they were granted credentials for their connection to different roles.

In the overall rating of the Web Portal, 81% of the users stated that they were very satisfied, and 16.7% said that they were satisfied, while 2.4% stated that their experience was neutral. There were no negative answers. In terms of ease of use, 76.2% of the users stated that they were very satisfied, and the remaining 23.8% said that they were satisfied. The processing speed was quite satisfactory, with 95.2% of the users stating that they were very satisfied, 2.4% that they were satisfied, and the remaining 2.4% stating that their experience was neutral. Finally, it was very important that all users easily understood the functions of the Web Portal, completed all their requests, and did not encounter any technical difficulties.

In Section 5, we provide an example scenario which shows how the ontology is used in real practice.

5. The Web Portal

The purpose of the portal is to provide stage 1 to 4 services to citizens and employees. The Web Portal manages the user interface and the communication with the SPARQL Endpoint. Users may log in and submit applications to the administration. They can also access their personal data through the personal repository. At the same time, we developed the relevant interfaces (Organization's Dashboard) for data management on the part of the administration. The e-government's stages of services in the web portal are the following:

- **Stage 1 (Information).** The web portal contains basic citizen information services. In general, providing information at this stage of services is not very complex. What is interesting is the data retrieval that is performed with SPARQL queries to the Apache Jena Endpoint. From the options menu, users can select the type of information that can be displayed.
- **Stage 2 (Interaction).** An additional feature offered through the second stage services is the one-way interaction. Users have the opportunity to interact with the portal by downloading documents and information which they can store on their computer and use for their physical transaction with the organizations. For the services of this stage, there is no communication with the SPARQL Endpoint and all management is accomplished through the website.
- **Stage 3 (Two-way Interaction).** The Web Portal supports two-way interaction services by offering to user's registration functions. Users can register by providing basic personal information and after the administration's approval, they have access to more online services. After their successful registration, they can submit applications to the administration. For submitting applications communication is necessary with the SPARQL Endpoint for retrieval of the necessary parameters.
- **Stage 4 (Transaction).** Stage 4 services are provided through the user's personal repository. The personal repository contains the employee or teacher's electronic file. The user has access to all the data kept by the administration in the file of the organization. He also has access to his request history and can check the status of his applications. Besides, he can download the certificates and decisions for which he has applied onto his computer.

5.1. Web Portal Functions

Some of the functions provided by the Web Portal are available to users without logging in, while others require a login. In Figure 5, we can see the options provided by the main menu to users.

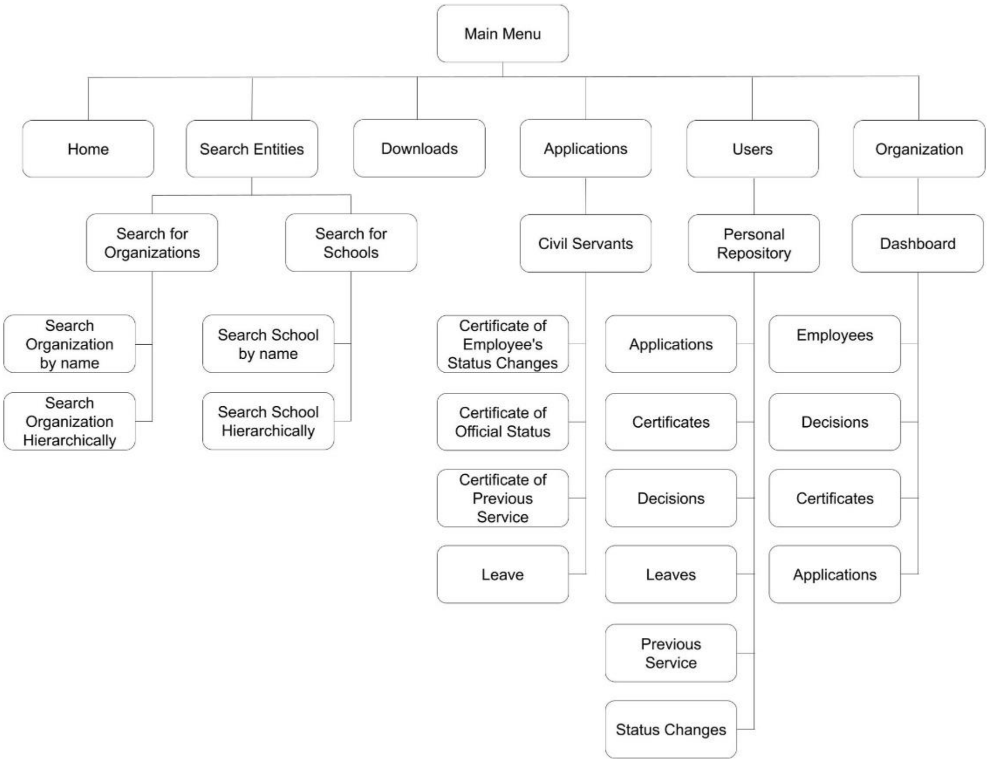


Figure 5. The services offered by the web portal.

5.2. *Life Cycle of a Request*

Here, we will briefly describe the life cycle of a request, from the moment the user starts the application process, until the receipt of the requested document. The procedure we will describe is the issuance of a Certificate of Employee's Status Changes. This procedure belongs to the "Application and automatic issuance of a document" functions. For the presentation, we split the procedure into two parts—the application submission and the preview of the certificate. As the task is fully automated, we do not separate the application from the issuance of the certificate. In Figure 6 we can see the classes, the object properties, and the data properties that we have to use in order to produce the Certificate of Employee's Status Changes.

5.2.1. Submission of the Application

To submit the application, the Web Portal communicates with the SPARQL Endpoint in two phases. In the first phase, we enter as soon as the user selects an application from the main menu, so it appears his intention to apply. At this stage the application has not been submitted yet and the necessary information for its completion is provided. In Figure 7 we see the relevant query.

We enter the second phase when the user clicks on the submit button. After that, the request of Figure 8 is submitted to the SPARQL Endpoint. For issuing the certificate, no action of the administration is required as the process is fully automated. Therefore, with this request the details of the application (lines 7–17) and the details of the certificate (lines 19–34) are inserted.

5.2.2. Preview of the Certificate

The issued (outgoing) documents are not static documents stored in the server storage. Every time a user requests to preview them, the Web Portal creates them dynamically by collecting data with queries submitted to the SPARQL Endpoint.

Once the user requests the preview of a certificate, a series of SPARQL queries are executed which collect the necessary data for constructing the document. If the queries are executed successfully, the Web Portal compiles the document and displays it to the user. These queries will provide us with information about the employee, the organization, and the document. The basis for all the queries is the registry number of the employee, except the queries for the documents that are based on the reference number. In Figures 9 and 10 we see two of these queries. The first one requests information about the employee and the second one information about the employee's status changes.

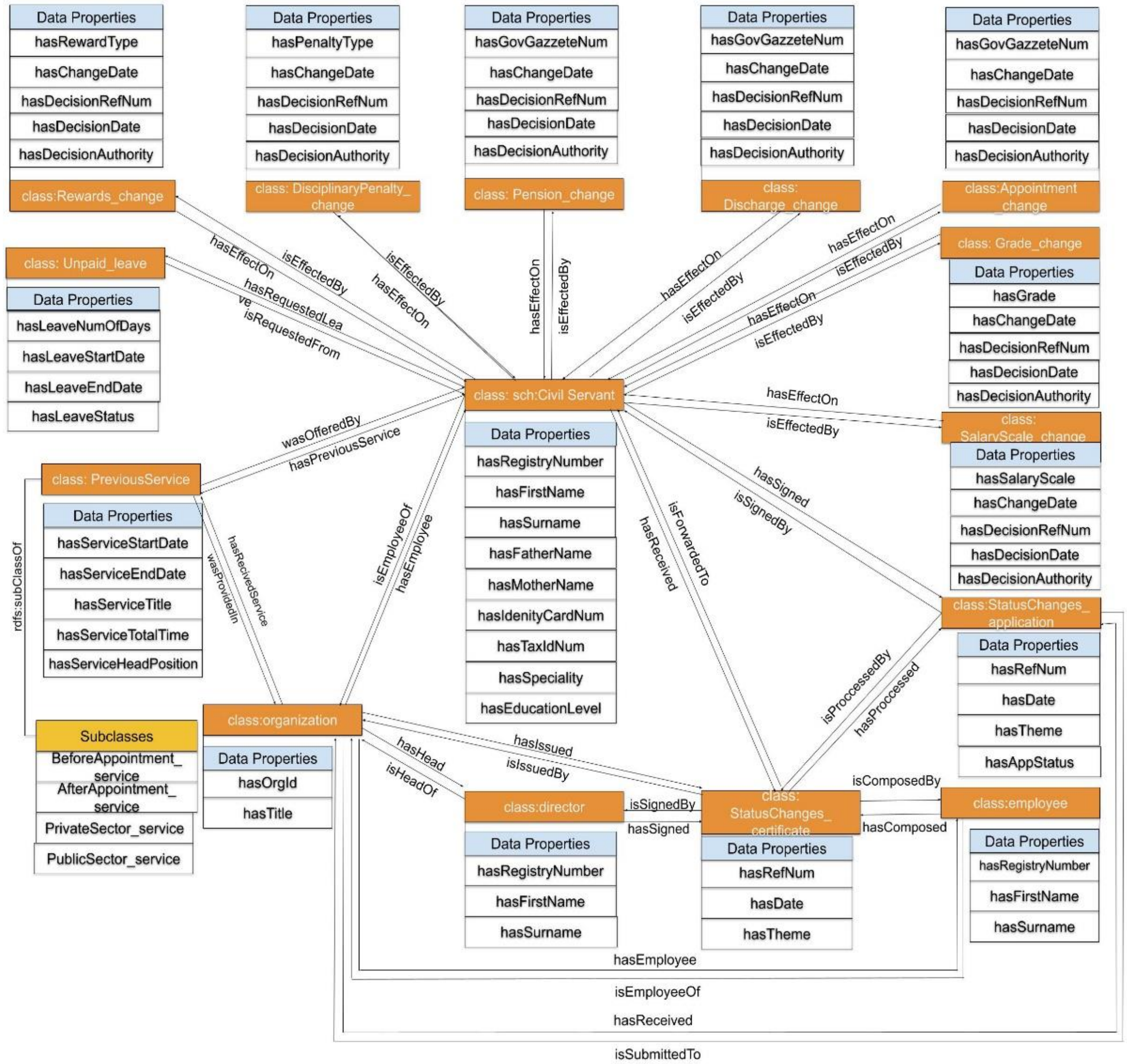
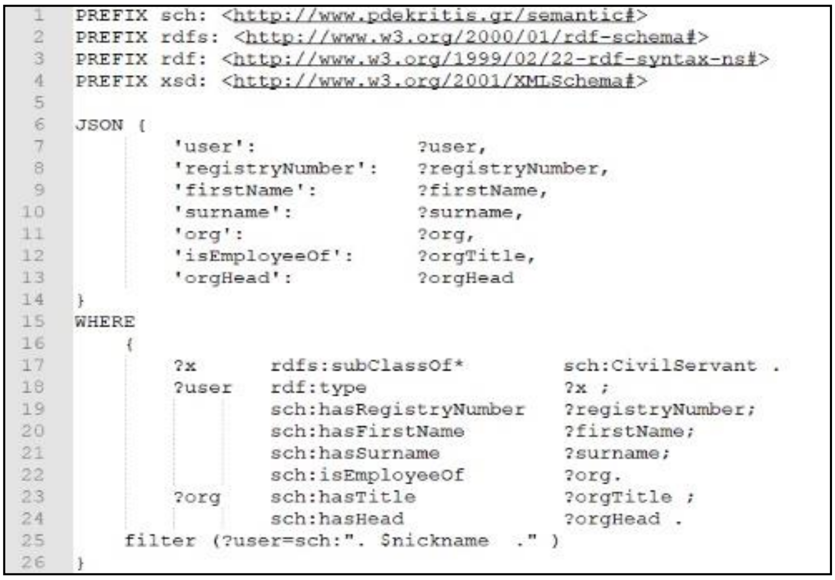
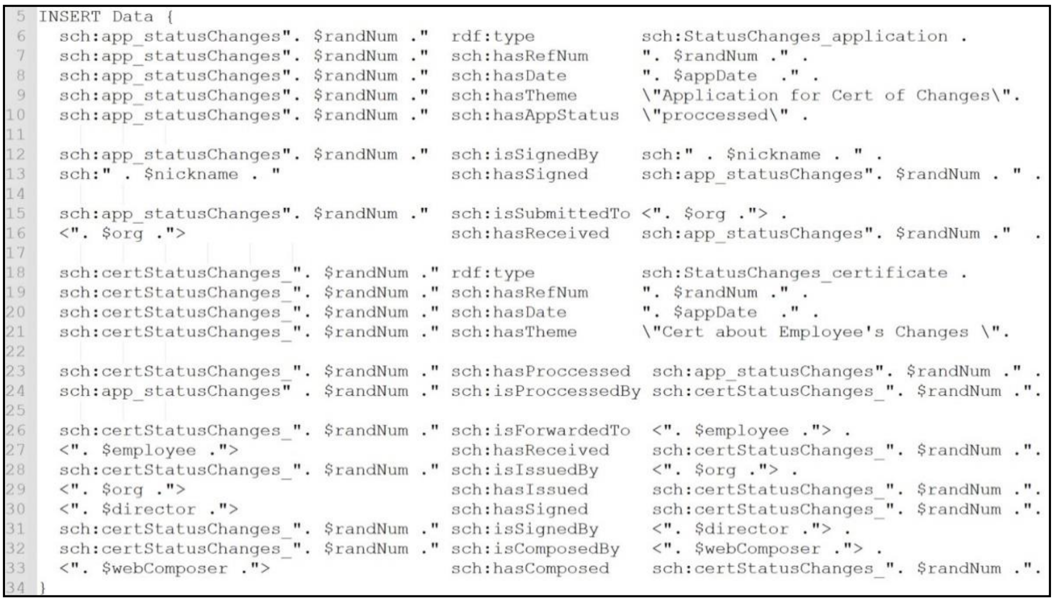


Figure 6. Classes, Object and Data Properties involved in issuing the Certificate of Employee's Status Changes.



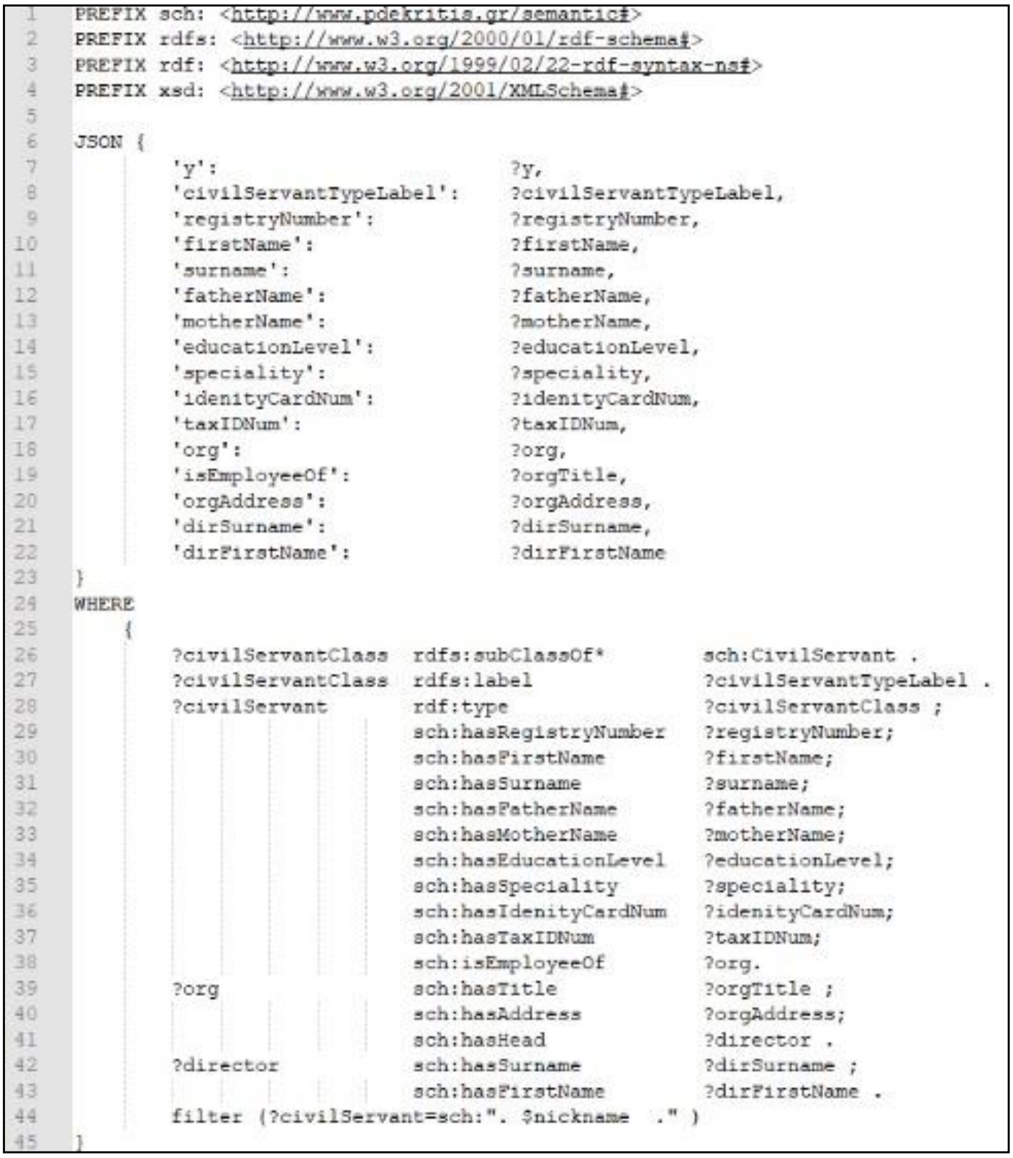
```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'user':              ?user,
        'registryNumber':    ?registryNumber,
        'firstName':         ?firstName,
        'surname':           ?surname,
        'org':               ?org,
        'isEmployeeOf':      ?orgTitle,
        'orgHead':           ?orgHead
}
WHERE
    {
        ?x      rdfs:subClassOf*        sch:CivilServant .
        ?user   rdf:type                ?x ;
                sch:hasRegistryNumber   ?registryNumber;
                sch:hasFirstName        ?firstName;
                sch:hasSurname          ?surname;
                sch:isEmployeeOf        ?org.
        ?org    sch:hasTitle            ?orgTitle ;
                sch:hasHead             ?orgHead .
    filter (?user=sch:". $nickname  ." )
}
```

Figure 7. SPARQL query for retrieving employee's and organization's information.



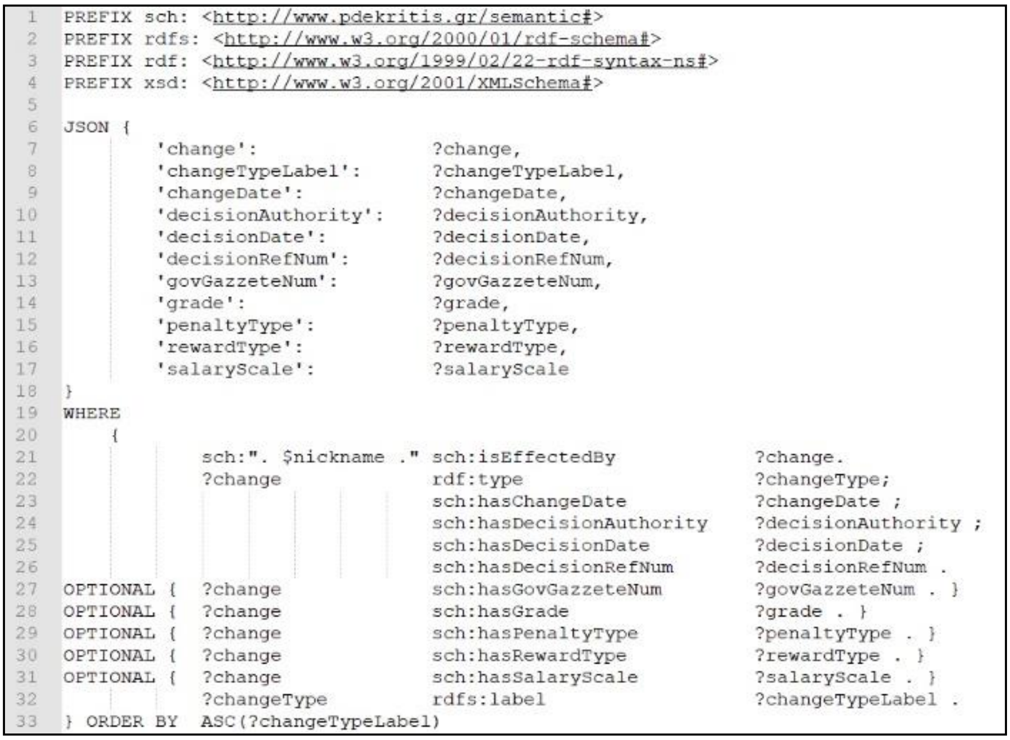
```
INSERT Data {
  sch:app_statusChanges". $randNum ."  rdf:type          sch:StatusChanges_application .
  sch:app_statusChanges". $randNum ."  sch:hasRefNum     ". $randNum ." .
  sch:app_statusChanges". $randNum ."  sch:hasDate       ". $appDate  ." .
  sch:app_statusChanges". $randNum ."  sch:hasTheme      \"Application for Cert of Changes\".
  sch:app_statusChanges". $randNum ."  sch:hasAppStatus  \"proccessed\" .

  sch:app_statusChanges". $randNum ."  sch:isSignedBy    sch:" . $nickname . " .
  sch:" . $nickname . "                sch:hasSigned     sch:app_statusChanges". $randNum . " .

  sch:app_statusChanges". $randNum ."  sch:isSubmittedTo <". $org ."> .
  <". $org .">                         sch:hasReceived   sch:app_statusChanges". $randNum ."  .

  sch:certStatusChanges_". $randNum ." rdf:type          sch:StatusChanges_certificate .
  sch:certStatusChanges_". $randNum ." sch:hasRefNum     ". $randNum ." .
  sch:certStatusChanges_". $randNum ." sch:hasDate       ". $appDate  ." .
  sch:certStatusChanges_". $randNum ." sch:hasTheme      \"Cert about Employee's Changes \".

  sch:certStatusChanges_". $randNum ." sch:hasProccessed  sch:app_statusChanges". $randNum ." .
  sch:app_statusChanges" . $randNum ." sch:isProccessedBy sch:certStatusChanges_". $randNum .".

  sch:certStatusChanges_". $randNum ." sch:isForwardedTo  <". $employee ."> .
  <". $employee .">                    sch:hasReceived    sch:certStatusChanges_". $randNum .".
  sch:certStatusChanges_". $randNum ." sch:isIssuedBy     <". $org ."> .
  <". $org .">                         sch:hasIssued      sch:certStatusChanges_". $randNum .".
  <". $director .">                    sch:hasSigned      sch:certStatusChanges_". $randNum .".
  sch:certStatusChanges_". $randNum ." sch:isSignedBy     <". $director ."> .
  sch:certStatusChanges_". $randNum ." sch:isComposedBy   <". $webComposer ."> .
  <". $webComposer .">                 sch:hasComposed    sch:certStatusChanges_". $randNum .".
}
```

Figure 8. SPARQL for registering Application and issuing Certificate.



```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'y':                        ?y,
        'civilServantTypeLabel':    ?civilServantTypeLabel,
        'registryNumber':           ?registryNumber,
        'firstName':                ?firstName,
        'surname':                  ?surname,
        'fatherName':               ?fatherName,
        'motherName':               ?motherName,
        'educationLevel':           ?educationLevel,
        'speciality':               ?speciality,
        'idenityCardNum':           ?idenityCardNum,
        'taxIDNum':                 ?taxIDNum,
        'org':                      ?org,
        'isEmployeeOf':             ?orgTitle,
        'orgAddress':               ?orgAddress,
        'dirSurname':               ?dirSurname,
        'dirFirstName':             ?dirFirstName
}
WHERE
    {
        ?civilServantClass  rdfs:subClassOf*       sch:CivilServant .
        ?civilServantClass  rdfs:label             ?civilServantTypeLabel .
        ?civilServant       rdf:type               ?civilServantClass ;
                            sch:hasRegistryNumber  ?registryNumber;
                            sch:hasFirstName       ?firstName;
                            sch:hasSurname         ?surname;
                            sch:hasFatherName      ?fatherName;
                            sch:hasMotherName      ?motherName;
                            sch:hasEducationLevel  ?educationLevel;
                            sch:hasSpeciality      ?speciality;
                            sch:hasIdenityCardNum  ?idenityCardNum;
                            sch:hasTaxIDNum        ?taxIDNum;
                            sch:isEmployeeOf       ?org.
        ?org                sch:hasTitle           ?orgTitle ;
                            sch:hasAddress         ?orgAddress;
                            sch:hasHead            ?director .
        ?director           sch:hasSurname         ?dirSurname ;
                            sch:hasFirstName       ?dirFirstName .
        filter (?civilServant=sch:". $nickname  ." )
}
```

Figure 9. SPARQL query for retrieving Employee’s information.



```
PREFIX sch: <http://www.pdekritis.gr/semantic#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

JSON {
        'change':                ?change,
        'changeTypeLabel':       ?changeTypeLabel,
        'changeDate':            ?changeDate,
        'decisionAuthority':     ?decisionAuthority,
        'decisionDate':          ?decisionDate,
        'decisionRefNum':        ?decisionRefNum,
        'govGazzeteNum':         ?govGazzeteNum,
        'grade':                 ?grade,
        'penaltyType':           ?penaltyType,
        'rewardType':            ?rewardType,
        'salaryScale':           ?salaryScale
}
WHERE
    {
            sch:". $nickname ." sch:isEffectedBy            ?change.
            ?change             rdf:type                    ?changeType;
                                sch:hasChangeDate           ?changeDate ;
                                sch:hasDecisionAuthority    ?decisionAuthority ;
                                sch:hasDecisionDate         ?decisionDate ;
                                sch:hasDecisionRefNum       ?decisionRefNum .
OPTIONAL {  ?change             sch:hasGovGazzeteNum        ?govGazzeteNum . }
OPTIONAL {  ?change             sch:hasGrade                ?grade . }
OPTIONAL {  ?change             sch:hasPenaltyType          ?penaltyType . }
OPTIONAL {  ?change             sch:hasRewardType           ?rewardType . }
OPTIONAL {  ?change             sch:hasSalaryScale          ?salaryScale . }
            ?changeType         rdfs:label                  ?changeTypeLabel .
} ORDER BY  ASC(?changeTypeLabel)
```

Figure 10. SPARQL query for retrieving Employee's changes.

6. Conclusions

In recent decades, progress in the field of information technology and communications has given a huge boost to the evolution of society, giving it an additional digital dimension. In this cycle of change, governments are called upon to meet the needs of society and to transfer their activities to the digital world through e-government. In this context, many government organizations have established online presence points offering online services to citizens, businesses, and other government bodies. However, the services they offer extend mainly to stages 1 and 2 and less to stages 3 and 4. Additionally, few organizations have been involved in providing online services to their employees.

Various solutions have been adopted for the development of online services. A very useful and effective tool for the development of such services are the ontologies. Ontologies offer better modeling of the structure of the entities, services, and data involved in the processes. They are also more effective in tackling heterogeneity problems. Moreover, the use of ontologies has the advantage of inference ability through reasoning.

At the same time, governments around the world, recognizing the need for e-government, have supported several initiatives and research projects in this area. These efforts have helped to lay out general frameworks. However, this task is quite complex and encounters many difficulties, as the scope of the public sector is huge. As a result, governments find it difficult to adopt the horizontal ontologies that will be the backbone for the development of vertical ontologies.

The solution we propose to address the above issues is based on the development of vertical ontologies to serve the specific needs of each organization. The approach that we propose for the development of e-government systems certainly raises issues of heterogeneity. However, given the difficulty in implementing strict frameworks for e-government, this is a more general problem that does not occur only in our case. In addition, the modeling offered by the use of ontologies helps us to overcome the issues of heterogeneity more easily.

In order to investigate the effectiveness of using ontologies to offer stages 1–4 services to employees, we developed an ontology and a Web Portal that support the administrative procedures of an organization. The ontology was designed from scratch, using a bottom-up approach, and was customized for the special needs of the organization. The Web Portal undertook to provide the services of stages 1–4, providing the necessary interfaces to the users and processing the necessary communication with the SPARQL Endpoint.

From the developer's point of view, the process of developing the services using the ontology was flexible and functional. Accessing the SPARQL Endpoint was a simple process that did not require complex queries. The evaluation of the users, after the use of the Web Portal, was positive. The users used all four stages of the services and successfully completed their requests without encountering any technical problems. Implementing services using ontologies was not something that was perceived by the users. The cooperation of the Web Portal with the SPARQL Endpoint was effective in meeting the needs of the users both for the employees and for the administration.

Our approach provides an adequate solution for the implementation of integrated e-government services, which can be effectively applied from government to employee and to the rest of the sectors of e-government.

Future Work

Our work is an initial approach to the development of vertical ontologies, which is open to a variety of extensions. Several issues can be explored at a later stage. First of all, there are plenty of administrative tasks that can be incorporated. Additionally, an issue that needs to be explored is the interconnection with other systems. The utilization of the third-party certification and specifically through the API of the General Secretariat of Information Systems will increase interoperability capabilities. Another issue on which there is already relevant research and which will expand the capabilities of the ontology is the incorporation of document templates. In addition, necessary features that need to be developed are the handling of the information provided in the context of information services as well as the handling of supporting incoming and outgoing documents. Moreover, our ontology can be extended to other areas of e-government, offering services to citizens and businesses, and further used for tasks of integrating data available already [42–44].

Author Contributions: Conceptualization, E.T. and N.P.; methodology, E.T.; software, E.T.; validation, E.T., H.K. and G.V.; writing—original draft preparation, E.T.; writing—review and editing, H.K.; visualization, E.T.; supervision, N.P.; project administration, N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Heeks, R.; Bailur, S. Analyzing e-government research: Perspectives, philosophies, theories, methods, and practice. *Gov. Inf. Q.* **2007**, *24*, 243–265. [CrossRef]
2. Ramaprasad, A.; Sánchez-Ortiz, A.; Syn, T. International e-government development: Policy, implementation and best practice. In *International E-Government Development: Policy, Implementation and Best Practice*; Muñoz, L.A., Bolívar, M.P.R., Eds.; Palgrave Macmillan: Cham, Switzerland, 2017; pp. 19–38, ISBN 9783319632841.
3. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. *Sci. Am.* **2001**, *284*, 34–43. [CrossRef]
4. Pileggi, S.F.; Fernandez-Llatas, C.; Traver, V. When the Social Meets the Semantic: Social Semantic Web or Web 2.5. *Future Internet* **2012**, *4*, 852–864. [CrossRef]
5. Mikroyannidis, A. Toward a social semantic web. *Computer* **2007**, 113–115. [CrossRef]
6. Munir, K.; Sheraz Anjum, M. The use of ontologies for effective knowledge modelling and information retrieval. *Appl. Comput. Inform.* **2018**, *14*, 116–126. [CrossRef]
7. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data—The story so far. *Int. J. Semant. Web Inf. Syst.* **2009**, *5*, 1–22. [CrossRef]
8. Guarino, N. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum. Comput. Stud.* **1995**, *43*, 625–640. [CrossRef]

9. Fraser, J.; Adams, N.; Macintosh, A.; McKay-Hubbard, A.; Lobo, T.P.; Pardo, P.F.; Martínez, R.C.; Vallecillo, J.S. Knowledge management applied to e-government services: The use of an ontology. In Proceedings of the IFIP International Working Conference on Knowledge Management in Electronic Government, Rhodes, Greece, 26–28 May 2003; Volume 2645, pp. 116–126. [CrossRef]
10. Apostolou, D.; Stojanovic, L.; Lobo, T.P.; Miró, J.C.; Papadakis, A. Configuring e-government services using ontologies. In Proceedings of the IFIP International Federation for Information Processing, Beijing, China, 7 February 2005; Springer: Boston, MA, USA, 2005; Volume 189, pp. 141–155.
11. Stojanovic, L.; Stojanovic, N.; Apostolou, D. Change management in e-government: OntoGov case study. *Electron. Gov.* **2006**, *3*, 74–92. [CrossRef]
12. Loukis, E.N. An ontology for G2G collaboration in public policy making, implementation and evaluation. *Artif. Intell. Law* **2007**, *15*, 19–48. [CrossRef]
13. Peristeras, V.; Tarabanis, K. Governance Enterprise Architecture (GEA): Domain models for e-governance. In Proceedings of the 6th International Conference on Electronic Commerce, Delft, The Netherlands, 25–27 October 2004; ACM: New York, NY, USA, 2004; Volume 60, pp. 471–479.
14. Goudos, S.K.; Peristeras, V.; Loutas, N.; Tarabanis, K. A public administration domain ontology for semantic discovery of eGovernment services. In Proceedings of the 2007 2nd International Conference on Digital Information Management, ICDIM, Lyon, France, 11–13 December 2007; IEEE: New York, NY, USA, 2007; Volume 1, pp. 260–265.
15. Xia, W.; Vitvar, T.; Peristeras, V.; Mocan, A.; Goudos, S.K.; Tarabanis, K. WSMO-PA: Formal specification of public administration service model on semantic web service ontology. In Proceedings of the 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07), Waikoloa, HI, USA, 3–6 January 2007; IEEE: Waikoloa, HI, USA, 2007; p. 96.
16. Shehu, J.; Xhina, E. Modeling an ontology for public E-Government Services in Albania. In Proceedings of the International Academic Conference on Research in Engineering and Technology, Paris, France, 9–11 July 2019; Diamond Scientific Publishing: Barcelona, Spain, 2019.
17. Salhofer, P.; Stadlhofer, B.; Tretter, G. Ontology driven e-government. *Electron. J. e-Gov.* **2009**, *7*, 415–424.
18. Alshehab, A.; Naser, N.A.; Hawaf, A.A. Semantic Integration Sharing for E-Government Domains Ontology: Design and Implementation Using OWL. *J. Theor. Appl. Inf. Technol.* **2019**, *97*, 1820–1831.
19. Kumar, A.; Joshi, A. IndiGov-O: An ontology of Indian government to empower digital governance. In Proceedings of the India International Conference on Information Processing, IICIP 2016, Delhi, India, 12–14 August 2016; IEEE: Delhi, India, 2017; pp. 4–9.
20. Gómez-Pérez, A.; Ortiz-Rodríguez, F.; Villazón-Terrazas, B. Legal ontologies for the Spanish e-Government. In Proceedings of the 11th Conference of the Spanish Association for Artificial Intelligence, Santiago de Compostela, Spain, 16–18 November 2005; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4177 LNAI, pp. 301–310.
21. Molnár, B.; Béleczki, A.; Benczúr, A. Application of legal ontologies based approaches for procedural side of public administration: A case study in Hungary. In Proceedings of the 5th International Conference on Electronic Government and the Information Systems Perspective EGOVIS 2016, Porto, Portugal, 5–8 September 2016; Springer: Porto, Portugal, 2016; Volume 9831 LNCS, pp. 135–149.
22. Savvas, I.; Bassiliades, N. A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration. *Expert Syst. Appl.* **2009**, *36*, 4467–4478. [CrossRef]
23. Hasan, M.M.; Anagnostopoulos, D.; Kousiouris, G.; Stamati, T.; Loucopoulos, P.; Nikolaidou, M. An Ontology based framework for e-Government regulatory requirements compliance. *Int. J. E-Serv. Mob. Appl.* **2019**, *11*, 22–42. [CrossRef]
24. Brusa, G.; Caliusco, M.L.; Chiotti, O. Building ontology in public administration: A case study. In Proceedings of the First International Workshop on Applications and Business Aspects of the Semantic Web, Athens, GA, USA, 26 December 2006; Volume 226, pp. 16–30.
25. Muñoz-Soro, J.F.; EstebAn, G.; Corcho, O.; Serón, F. PPROC, an ontology for transparency in'public procurement. *Semant. Web* **2016**, *7*, 295–309. [CrossRef]
26. Lourenço, R.P.; Jorge, S.; Rolas, H. Towards a transparency ontology in the context of open government. *Electron. Gov. Int. J.* **2016**, *12*, 375–394. [CrossRef]
27. Ding, L.; Lebo, T.; Erickson, J.S.; Difranzo, D.; Williams, G.T.; Li, X.; Michaelis, J.; Graves, A.; Zheng, J.G.; Shangguan, Z.; et al. TWC LOGD: A portal for linked open government data ecosystems. *J. Web Semant.* **2011**, *9*, 325–333. [CrossRef]
28. Gugliotta, A.; Cabral, L.; Domingue, J.; Roberto, V.; Rowlatt, M.; Davies, R. A semantic web service-based architecture for the interoperability of e-government services. In Proceedings of the Web Information Systems Modeling Workshop (WISM 2005)/5th International Conference on Web Engineering (ICWE 2005), Porto, Portugal, 13–17 June 2005; pp. 21–30.
29. Xiao, Y.; Xiao, M.; Zhao, H. An ontology for e-government knowledge modeling and interoperability. In Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007, Shanghai, China, 21–25 September 2007; IEEE: Shanghai, China, 2007; pp. 3605–3608.
30. Kanagwa, B.; Nakatumba-Nabende, J.; Mugwanya, R.; Kigozi Kahiigi, E.; Ngabirano, S. Towards an Interoperability e-Government Framework for Uganda. In Proceedings of the 9th International Conference, AFRICOMM 2017, Lagos, NG, USA, 11–12 December 2017; Springer: Cham, Switzerland; Lagos, NG, USA, 2018; pp. 16–28.

31. Asfand-E-Yar, M.; Ali, R. Semantic integration of heterogeneous databases of same domain using ontology. *IEEE Access* **2020**, *8*, 77903–77919. [CrossRef]
32. Chbihi Louhdi, M.R.; Behja, H. Ontology learning from relational databases: Transforming recursive relationships to OWL2 components. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 265–270. [CrossRef]
33. Hazber, M.A.G.; Li, R.; Gu, X.; Xu, G. Integration mapping rules: Transforming relational database to semantic web ontology. *Appl. Math. Inf. Sci.* **2016**, *10*, 881–901. [CrossRef]
34. Tissot, H.; Huve, C.A.G.; Peres, L.M.; Fabro, M.D. Del Exploring logical and hierarchical information to map relational databases into ontologies. *Int. J. Metadata Semant. Ontol.* **2019**, *13*, 191–208. [CrossRef]
35. Musen, M.A.; Protégé Team. The Protégé Project: A Look Back and a Look Forward. *AI Matters* **2015**, *1*, 4–12. [CrossRef] [PubMed]
36. Sirin, E.; Parsia, B.; Grau, B.C.; Kalyanpur, A.; Katz, Y. Pellet: A practical OWL-DL reasoner. *Web Semant.* **2007**, *5*, 51–53. [CrossRef]
37. Poveda-Villalón, M.; Gómez-Pérez, A.; Suárez-Figueroa, M.C. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *Int. J. Semant. Web Inf. Syst.* **2014**, *10*, 7–34. [CrossRef]
38. Rogier, J.; Sampson, J.; Verhelst, F. Position Paper—W3C Workshop on Semantic Web in the Energy Industries; Part I: Oil & Gas. In Proceedings of the W3C Workshop on Semantic Web in Oil & Gas Industry, Houston, TX, USA, 9–10 December 2008; Volume 1, pp. 1–4.
39. Francesconi, E.; Montemagni, S.; Peters, W.; Tiscornia, D. Integrating a bottom-up and top-down methodology for building semantic resources for the multilingual legal domain. In *Semantic Processing of Legal Texts*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6036, pp. 95–121. [CrossRef]
40. Uschold, M.; Gruninger, M. ONTOLOGIES: Principle, Methods and Applications. *Knowl. Eng. Rev.* **1996**, *11*, 93–136. [CrossRef]
41. Vassilakis, C.; Lepouras, G. Ontology for E-Government Public Services. In *Theory and Applications of Ontology: Computer Applications*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 865–870. [CrossRef]
42. Kondylakis, H.; Plexousakis, D. Ontology evolution: Assisting query migration. International Conference on Conceptual Modeling. *Int. Conf. Concept. Model.* **2012**, *7532*, 331–344. [CrossRef]
43. Kondylakis, H.; Plexousakis, D. Ontology evolution in data integration: Query rewriting to the rescue. International Conference on Conceptual Modeling. *Int. Conf. Concept. Model.* **2011**, *6998*, 393–401. [CrossRef]
44. Minadakis, N.; Marketakis, Y.; Kondylakis, H.; Flouris, G.; Theodoridou, M.; de Jong, G.; Doerr, M. X3ML Framework: An Effective Suite for Supporting Data Mappings. In Proceedings of the Extending, Mapping and Focusing the CRM, 19th International Conference on Theory and Practice of Digital Libraries (TPDL2015), Poznań, Poland, 14–18 September 2015; pp. 1–12.