An Ontology-based Conversational Governmental Portal

Nikolaos Loutas[a,b *], Vassilios Peristeras[b,c] and Konstantinos Tarabanis[b,d]
[a] *Information Systems Lab, University of Macedonia, Egnatia 156, 54006, Thessaloniki, Greece*
{nlout, kat}@uom.gr
[b] *Digital Enterprise Research Institute, National University of Ireland, Galway, IDA Business Park, Lower Dangan, Galway, Ireland*
firstname.lastname@deri.org
[c] *Greek National Center for Public Administration and Decentralization, Thessaloniki, Greece*
[d] *Centre for Research and Technology Hellas, 6th km Charilaou-Thermis rd., 57001, Thessaloniki, Greece*

Abstract. Public service provision is a complex process influenced by various factors. Public services have a complex structure and usually span across different public agencies. Hence, citizens are usually not aware of the way services have been organized within the administrative system. Moreover, the legislation that governs a public service is usually hard to understand. Governments worldwide have developed portals providing eGovernment services in order to simplify and facilitate public service provision. But most of these portals do not manage to hide from users the complexity of public services and they become themselves complex and difficult to use. At the same time they fail to address the specific needs of a particular user as the analysis of the services they usually make available is only enough for obtaining general information about the service. This work presents an ontology-based conversational governmental portal that helps citizens find the public services that address their individual needs. The portal uses a user-friendly, self-explanatory interface that supports the citizens and offers them the necessary guidance. Additionally, it consults on the eligibility of a citizen for receiving a particular public service and provides a rich, structured and personalized public service description. Finally, the citizen can use the portal in order to invoke the identified service, if available on-line.

Keywords: ontology, OWL, semantics, public service, portal, eGovernment, conversational

1. Introduction

The public service provision process can be broken down into two distinct but complementary parts: the *informative* and the *performative* part [1]. During the informative part, information is reciprocally communicated between citizens/businesses (in the rest of the paper we refer only to citizens for the sake of brevity but the discussion applies to businesses as well) and public administration (PA), while during the performative part certain actions (changes in the real world) occur.

Currently, these two parts are discussed together, but we prefer to analyze and model them separately. By doing so, more accurate understanding and modeling of each part becomes possible. In this paper, our focus is on the informative part. Interestingly, this part is usually overlooked by both professionals and scholars, although its organization and facilitation are critical in the overall service provision process. During the informative part citizens try to identify the public services that address their needs and find answers to a set of questions regarding these services, e.g. am I eligible for this service, what is the outcome of the service, which public agency provides the service, where is the service provided from, which are the required inputs, the preconditions etc.

Supporting the informative part becomes more difficult if the service versioning problem is considered. Public services are complex structures which are de-

* Corresponding Author. Email: nlout@uom.gr. Phone: +30 2310 891504. Fax: +30 2310 891509

composed into a, usually large, number of service versions. It is therefore not enough for citizens to find the public service that addresses their need, but they should also go one step further and identify the specific service version for which they are eligible among the various versions of the specific public service. The service versioning problem is come across in almost all public services. For instance, "Issuance of Driving License" or "Establish a new Business" are examples of public services which are decomposed into various service versions.

The service versions refine further the generic public service and differentiate from one another due to differences in:

i. the profile of the citizen that wishes to consume the service;
ii. the service inputs and outputs; and/or
iii. the service workflow.

Only the service versions can finally be executed and consumed by citizens. It is interesting that even for public services that are considered simple, like the "Issuance of Driving License", many different service versions exist e.g. Driving License for people over 60, Driving License for disabled people, Driving License for driving small or large vehicles etc.

Until now public administration provides minimum support to citizens during the informative part of service provision. Citizens usually act on an ad hoc basis and try to find their way to the desired public service and to acquire information about it following a trial and error approach. The service versioning problem is also not tackled by existing eGovernment portals, where the different characteristics, skills, preferences etc. of each citizen are not considered. Similarly, the eligibility check is currently not conducted during the informative part. This means that a citizen may go through a lengthy, time-consuming process in order to collect all the necessary inputs etc. for a public service, only to find out that he is finally not allowed to use it.

The complexity of the informative part of service provision is scaled even more by the fact that the descriptions of public services are fragmented and determined by different complementary laws and regulatory acts. Thus, in order to get a complete and comprehensive description of a public service these different, and often conflicting or difficult to understand, information has to be put together and homogenized. Until today this is only partially done and the information that is gathered is usually incomplete or not valid. Therefore, many of the questions that citizens have during the informative part of public service provision remain unanswered.

Hence, a clear requirement is derived for an infrastructure that will guide and will provide assistance to citizens throughout the informative part of public service provision. This infrastructure should help them find the specific public service version that they are eligible for and should make available all the service related information which is required for answering the questions that the citizens have during the informative part of service provision.

In light of this requirement, governments have spent significant amounts of effort and resources for improving the informative part of service provision, mainly through the development of eGovernment portals.

However, traditional governmental portals still follow a one-size-fits-all approach. Thus the portal cannot react differently and tailor the offered public services to the needs and the profile of each individual citizen. Moreover, the citizen has to figure out on his/her own whether s/he is eligible for the service by reading lengthy public service descriptions (which very often include legal terminology). These are common problems in all existing national eGovernment portals. According to [2], the most typical problems of eGovernment portals can be grouped into the following categories:

- The user is unable to find the desired information or service.
- The user is unable to achieve his goal, even though the system supports it and he has started along the path to achieve it.
- The user is able to accomplish his goal, but not efficiently, e.g., easily and quickly.

In order to enhance the informative part of public service provision and improve existing governmental portals, this paper introduces an ontology-based conversational portal (henceforth referred to as Portal) which aims:

- To inform citizens whether they are eligible for a specific public service;
- To personalize the public service related information according to the profile and the specific needs and wants of the citizen and identify the specific public service version;
- To provide complete and well-structured information for the public service;
- And (if a service execution environment is in place) to allow citizens to invoke public services that are available online.

The functionalities of the Portal are entirely based on the use of OWL ontologies. Hence, ontologies are used for:

i. supporting the structured dialogue between users and public administration ;
ii. modeling the users of the Portal;
iii. modeling public services; and
iv. representing public service related information (either collected and encoded in ontologies by the authors or by reusing ontologies created by others).

Fig. 1 shows the different types of ontologies used by the Portal and their independencies.

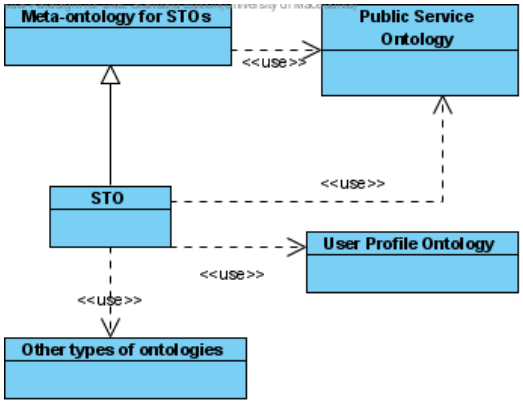

Fig. 1. Ontologies used by the Portal.

OWL was selected as it is constitutes a well accepted Semantic Web standard. OWL is a W3C recommendation since 2004 and according to a 2007 survey it is the most popular ontology language among Semantic Web experts [3]. Finally, it is relatively easy to translate between OWL and RDF/RDF-S, or OWL and WSML.

The Portal was initially developed in the context of the SemanticGov project[1] [4], where it played the role of the national Member State portal. It served as an entry point for the citizens to the public services offered by the SemanticGov platform. Two prototypes of the Portal were installed at the Region of Central Macedonia in Greece and the City of Turin in Italy [5]. Currently, the Portal is one of the three building blocks of the Rural Inclusion[2] platform. The Portal is used for supporting the provision of various public services in five trial sites located in four different EU countries, namely France, Greece, Latvia and Spain [6].

A running prototype of the Portal for testing purposes is available at http://www.semantic-gov.org/SemanticGovPortal/.

[1] http://www.semantic-gov.org
[2] http://www.rural-inclusion.eu

The rest of this paper is structured as follows. Section 2 discusses related efforts. The Portal capitalizes on OWL ontologies. The different types of ontologies used are discussed in sections 3, 4 and 5. The functionalities of the Portal and an overview of its architecture are given in section 6. Finally, our conclusions and future research directions are discussed in section 7.

2. Related Work

Researchers have tried to solve parts of the problem that analyzed in the previous section, focusing mostly on facilitating service search and discovery.

Fang et al. [7] support the selection of an optimal set of featured service-links. These links will then appear on the homepage of an eGovernment portal, thus helping users to locate services more easily by reducing the number of steps that they have to perform until the desired service is found. This is expected to improve the citizens' satisfaction and consequently increase the number of people using the portal. Therefore, a heuristic Web-mining algorithm called ServiceFinder is proposed, which aims to help citizens find the services that they are looking for in eGovernment portals. ServiceFinder uses three metrics to measure the quality of eGovernment service selection, which will then appear as links on the homepage of the portal. These are effectiveness (degree of easiness to locate the desired service), efficiency (probability to locate the desired service) and utilization (sum of easily located desired services). The metrics are calculated using patterns either extracted from the structure of the eGovernment portal or mined from a Web log.

Although this approach may improve the service discovery by organizing better the available services within the portal, the process of finding a service in the portal is still based on a trial and error approach. This means that the user is still browsing the eGovernment portal in order to find the desired service. Another drawback of this approach as compared to ours is that it provides no information to the citizen with respect to his/her eligibility for the identified public service.

Sacco [8] proposes a solution enabled by dynamic taxonomies, which support different facets that may be used by citizens. The facets that the system provides are: services, events of life, type of information, location, type of citizenship, person with special rights and person profile. The use of dynamic tax-

onomies makes this approach very flexible and fast. This is due to the fact that dynamic taxonomies adapt dynamically to the subset of the universe on which the user is focusing. The use of multiple facets enhances further the agility of the approach.

Nonetheless, this work suffers from similar problems as other approaches that organize services in hierarchical category trees. The user may have to browse deep into the dynamic taxonomy and should also be aware of the way that public administration has decided to organize services (even if multiple facets are made available), which may differ from his/her perspective. Therefore, the cognitive effort of citizens is not reduced as much as expected. The eligibility question remains unanswered in this case as well.

Recently, Stollberg and Muth [9] proposed an approach for service customization which is based on simplified service variants that expose only the service features that are relevant to the profile of a specific user or to the characteristics of a specific usage scenario. Hence, the authors define *(i)* the service variability metamodel, *(ii)* the main actors and *(iii)* the lifecycle of the service customization process. They also provide a set of tools based on the metamodel to support the service variant creation.

This work approaches the problem of service customization from a different perspective than the one implemented in our work. However, we too acknowledge the need to formalize the process. We therefore use the Public Service Ontology described in section 4 to model public services and the SBVR standard [12] for formally expressing the business rules that lead to different service versions.

The OneStopGov project[3] delivered a life-event portal that supports the active, citizen-centric approach [10]. The portal is based on a structured dialogue, which is based on workflows that model the internal workflow of a life-event, in order to personalize the life-event to the citizen's profile and to facilitate its execution. Hence, the OneStopGov approach adapts the life-event to citizen's profile, which practically means that citizens with different profiles will most likely execute different versions of the same life-event.

In the FIT project[4], an adaptive eGovernment portal has been implemented [11]. The approach, which employs Semantic Web and Web 2.0 technologies, proposes a framework which captures the user's behavior in the portal and adapts the portal accordingly. FIT's approach also recognizes the service versioning problem and tries to overcome this by providing personalized eGovernment services. In order to achieve this, the FIT portal follows an iterative ontology-driven adaptation approach (monitor, analyze, plan, execute). The FIT portal also uses OWL ontologies to model users, events and adaptation rules.

The Portal presented in this paper tries to fulfill similar objectives like the related efforts described so far. Nevertheless, some differences can be spotted both in terms of functionalities provided and in terms of the technologies used. For example, none of the related efforts decide on the eligibility of the citizen for a public service before the execution of the service. This is a very strong asset of our approach, as the eligibility check at an early stage during the informative part of service provision can save the citizen a lot of time and money.

It is interesting that the work of [8] bears some resemblance with ours in the way that services are modeled and organized, but what is different, apart from the use of ontologies versus taxonomies, is the fact that in our work services are described at a greater level of granularity e.g. the distinction between service type and service version. This difference is very important and due to this Sacco's work is not able to personalize services, but only provide generic info about them. Moreover, it does not answer the eligibility question.

This brings us also to the main difference between our work and the OneStopGov approach, where the citizen is not informed on their eligibility for a public service during the informative part but this is done only after the invocation of the service. Moreover, in the OneStopGov approach, they discuss about life-events while our work focus on services, thus the two efforts share similar objective but try to address them using a different level of granularity of the service, as well as different tools and technologies. Both approaches however focus on the informative part of service provision and as such are decoupled from the underlying service execution environments.

The personalization of the eGovernment services in FIT is based on tracking the user's behavior and adapting accordingly the forms that the user has to fill for a specific eGovernment service, while in our case eGovernment service personalization relies on the online dialogue between the user and the Portal

Finally, our Portal is based OWL ontologies. This enables us to offer to our users a dynamic, online structured dialogue, which guides them throughout the public service provision process. Moreover, ontologies allow us to dynamically generate the user

[3] www.onestopgov-project.org

[4] http://www.fit-project.org/

interface of the Portal and to make new services available through the Portal in a plug-n-play fashion. This means that as soon as a new ontology which models a service is stored in the Portal, the Portal can support the informative part of the provision of this service. All these are discussed in detail in the following sections.

3. Services Tree Ontologies

Currently, during the informative phase an informal dialogue between public administration and the citizen is established. During this dialogue, both the citizen and public administration set questions and provide answers in order to clarify if the citizen is eligible for the requested public service and, if eligible, to identify the specific service version. The questions set derive from the business rules that regulate the provision of a specific public service.

This work formalizes the dialogue that takes place during the informative phase of service provision using a tree-like structure. The dialogue starts from a generic public service which is stepwise refined after every question-answer pair. In case that the citizen is eligible for the specific public service, the dialogue leads to the public service version that matches his/her profile and a detailed structured description of the public service version is made available. Otherwise the citizen is informed that he/she is not allowed to use the specific public service. The dialogue is visually represented by means of a UML activity diagram.

In order to produce executable versions of the various dialogues, we encode them as OWL ontologies, called Service Tree Ontologies (STO). Hence, STOs capture and model the way a specific public service is decomposed into different service versions. They contain the business rules from which the different service versions derive as well as the questions that will be asked to the citizen in order to collect information that will allow the Portal to personalize the public service, decide on the eligibility of the citizen and on the matching service version. Moreover, the user interface of the Portal is dynamically created based on information encoded in the STOs.

In order to formalize the development of STOs, we have created a meta-ontology that defines the classes that comprise an STO. Each STO is then created as an instance of the meta-ontology.

The meta-ontology contains classes that have derived from the modeling of the aforementioned dialogue. Hence, the following classes have been included in the meta-ontology for STOs (see Fig. 2):

- ***Node.*** Nodes of an STO represent different states of the dialogue. A node has the following attributes:
  - The *hasDescription* attribute provides a brief description of the node, as to what the node represents in the STO.
  - The *hasEvidencePlaceholder* attribute refers to the administrative documents, e.g. certificates, which relate to a specific service version. A *Node* may contain zero or more *EvidencePlaceholders*.
  - The *containsPieceOfInformation* attribute refers to other types of information related to a specific service version, e.g. the amount of a fee that has to be paid. A *Node* may contain zero or more *PieceOfInformation*.

The following three concepts, i.e. *InternalNode* and *LeafNode* have been defined in the ontology as subclasses of *Node*. They thus inherit all its attributes.

- ***InternalNode.*** This class represents those nodes of the STO that have descendants. Apart from the attributes that they inherit from *Node*, *InternalNodes* have also:
  - The *hasChildNode* attribute which indicates the descendants of the current node. There can be more than one descendants, which constitute specializations of their parent node, mainly by containing more information about the citizen's profile.
  - The *hasQuestion* attribute which refers to a specific question asked to the citizen.
  - The *isRoot* attribute which indicates whether the specific node is the initial node of the dialogue or not.
- ***LeafNode.*** This class represents those nodes of the STO that have no descendants. *LeafNodes* indicate the termination of the dialogue, let it be successful or not. Apart from the attributes that they inherit from *Node*, *LeadNodes* have also the *isNotEligible* attribute which if true indicates that the citizen is not allowed to use the specific public service.
- ***Question.*** This class represents the questions that the Portal poses to the citizen. It has two attributes:
  - *hasData* models the question itself, e.g. "What is your marital status?"
  - *hasAnswer* models the possible answers, e.g. in the previous question "married, single, divorced, widow".

– ***SparqlQuery.*** This class represents formally a business rule of a specific public service. The business rule is expressed as a SPARQL query which is stored in the *hasData* attribute.

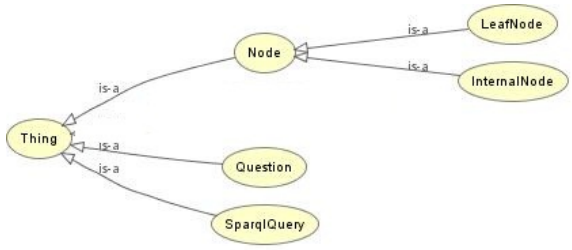

Fig. 2. The meta-ontology for STOs.

3.1. STO example: The Registration in the Chamber of Commerce public service

This section discusses how an STO for the case of the Registration in the Chamber of Commerce was developed. This public service is offered by all Chambers of Commerce in Greece and is obligatory for specific types of enterprises and entrepreneurs that are active in the area of jurisdiction of a specific Chamber of Commerce.

Simple as it may sound, this public service still has more than 20 different versions which stem from variations in the legal status of the enterprise/entrepreneur, on the type of their activity and on their nationality.

A documentation exercise was carried out, in order to gather all the details with regards to the provision of the specific service. Hence, the business rules that identify the service provision were identified and the activity diagram of the structured dialogue was designed. This was afterwards encoded in an OWL ontology.

For example, one of the business rules of the specific public service expressed in structured English (following SBVR [12]) reads “It is obligatory that each SA company with equity capital less than 64,000 Euros pays an annual fee of 160 Euros.” In order to collect the information that validates this rule, a question was created asking the equity capital of the SA company, i.e. “Please provide the equity capital of your company”.

In the STO developed for the Registration in the Chamber of Commerce public service, this question was then attached to one of the *InternalNodes*. This specific *InternalNode* will have a number of descendants, in our case three. One of them will correspond to the case of an SA company with capital less than 64,000 Euros. This will be expressed by a SPARQL expression linked to this specific node. The OWL code of the ontology instances described so far is presented in Table 1.

Table 1.

Sample from the Registration in the Chamber of Commerce STO.

The InternalNode instance

```
<owl:Thing rdf:about="#ObligesRegistration_SA">
        <rdf:type
rdf:resource="&www;PortalOntology.owl#InternalNode"/>
        <prtl:containsPieceOfInformation
rdf:datatype="&xsd;string" >Registration fee
160</prtl:containsPieceOfInformation>
        <prtl:hasEvidencePlaceholder
rdf:resource="&www;DocumentOntology.owl#Application"/>
        <prtl:hasEvidencePlaceholder rdf:resource=
"&www;DocumentOntology.owl#ArticleOfIncorporation"/>
        <prtl:hasEvidencePlaceholder rdf:resource="
&www;DocumentOntology.owl#BusinessInceptionCertificate"/>
        <prtl:hasEvidencePlaceholder
rdf:resource="&www;DocumentOntology.owl#Chios_Perfecture_
Statement"/>
        <prtl:hasEvidencePlaceholder
rdf:resource="&www;DocumentOntology.owl#IDorPassport"/>
        <prtl:hasCondition
rdf:resource="#LicencedSACompany"/>
        <prtl:hasChildNode rdf:resource="#SA_equitybetween"/>
        <prtl:hasChildNode rdf:resource="#SA_equityless"/>
        <prtl:hasChildNode rdf:resource="#SA_equitymore"/>
        <prtl:hasQuestion rdf:resource="#qstnEquityCapital"/>
    </owl:Thing>
```

The Question instance

```
  <owl:Thing rdf:about="#qstnEquityCapital">
        <rdf:type
rdf:resource="&www;PortalOntology.owl#Question"/>
        <prtl:hasAnswer
rdf:datatype="&xsd;string"></prtl:hasAnswer>
        <prtl:hasData xml:lang="en"> Please provide the equity
capital of your company </prtl:hasData>
    </owl:Thing>
```

The SparqlQuery instance

```
  <owl:Thing rdf:about="#EquityLess">
        <rdf:type
rdf:resource="&www;PortalOntology.owl#SparqlQuery"/>
        <rdfs:comment rdf:datatype="&xsd;string"
          >Checks if the EU applicant requires licensing and is
registering an association</rdfs:comment>
        <prtl:hasData rdf:datatype="&xsd;string"
          >PREFIX bo:&lt;http://www.owl-
ontologies.com/BusinessOntology.owl#&gt;          PREFIX
rdf:&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;
PREFIX xsd:&lt;http://www.w3.org/2001/XMLSchema#&gt;
SELECT ?x FROM &lt;http://www.owl-
ontologies.com/BusinessOntology.owl&gt;          WHERE { ?x
rdf:type ?t.  FILTER(?t=bo:SME). ?x bo:hasEquityCapital ?it.
FILTER(?it=bo:Fund_less_Than_64000).}</prtl:hasData>
    </owl:Thing>
```

4. Public Service Ontology

The Portal capitalizes on the GEA Public Service Model implementation in OWL (Public Service Ontology) [13]. The Public Service Ontology, which is depicted in Fig. 3, is used for representing public service related information by creating instances of the various classes that comprise the description of the public service.

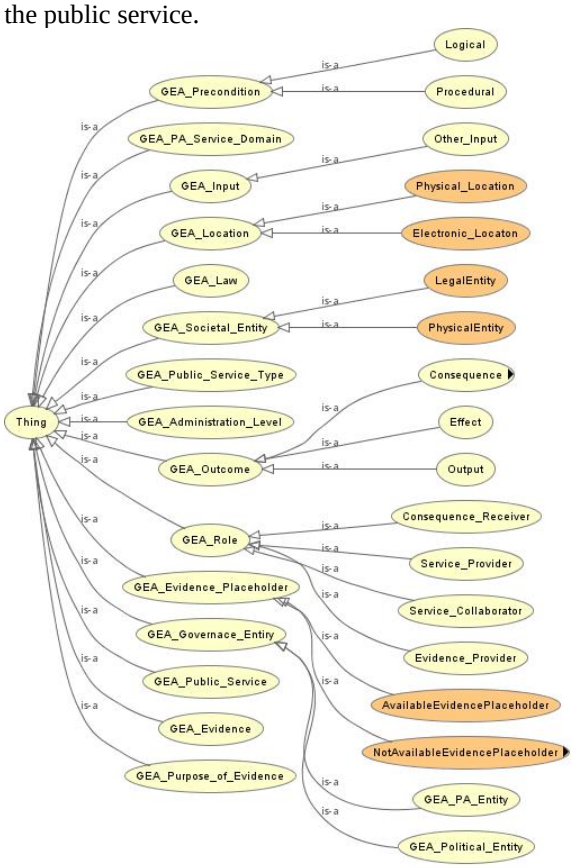

Fig. 3. The Public Service Ontology.

A brief description of the Public Service Ontology's classes follows. *Societal Entities* (e.g. citizen, business) have *Needs* related to specific *Goals*. A *Societal Entity* requests a *Public Administration (PA) Service* to serve its *Goals*. PA Services are categorized in several *Domains* (e.g. Health, Transportation). Each *Domain* object is divided into several *SubDomain* objects (e.g. Domain Transportation has SubDomains Ground Transportation, Air Transportation and Water Transportation).

There are several types of *Social Entities* (e.g. legal entity, physical person). There are two categories of *Governance Entities* participating in service provision: *Political* Entities and *Public Administration Entities*. *PA Entities* can acquire one of the following four *Roles* during the service execution phase:

- *Service Provider* is the PA Entity that provides the service to the Societal Entities (clients). The PA Entities belong to an *Administrative Level* (e.g. municipality, regional).
- *Evidence Provider* is the PA Entity that provides necessary Evidence to the Service Provider in order to execute the PA Service.
- *Consequence Receiver* is the PA Entity that should be informed about a PA Service execution.
- *Service Collaborator* is the PA Entity that participates in the provision of a public service (but is not the service provider).

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

Preconditions are validated by *Piece of Evidence* serving a *Purpose*. As *Evidence* is primarily pure information, it is stored in *Evidence Placeholders*, thus the *Evidence Placeholder* contains *Pieces of Evidences*. *PA Service* use specific types of *Evidence Placeholders* as *Input*.

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

- *Output*, which is the documented decision of the *Service Provider* regarding the service asked by a *Societal Entity*. This is currently embedded and reaches the client in the form of an *Evidence Placeholder*.
- *Effect*, which is the change in the state of the real world (e.g. transfer money to an account) caused by the execution of a service. In cases where administration refuses the provision of a service, there is no Effect.
- *Consequence*, which is information about the executed *PA Service* that needs to be forwarded to interested parties.

5. Other ontologies

Apart from the meta-ontology for STOs, the STOs and the Public service ontology, the Portal uses other types of OWL ontologies as well, such as:

– *Ontologies that model the profile of businesses and citizens*. For example, in the case of the Registration in the Chamber of Commerce public service described in section 3.1, a business ontology was created out, which represents the various characteristics of a business, e.g. brand name, type and legal status.
– *Ontologies that contain public service related information*. For example, in the case of the Registration in the Chamber of Commerce public service described in section 3.1, an ontology was developed that contains all the administrative documents (modeled as instances of the EvidencePlaceholder class of the Public Service Ontology) that are required as input for the different versions of the public service.
– *Ontologies that include listings of countries, nationalities, business types*. For example, in the case of the Registration in the Chamber of Commerce public service described in section 3.1, an ontology was used that contains a list of all different business types/activities, e.g. hotel, bakery, shop etc.

6. Architecture

The Portal follows a simple three-tier architecture that comprises of:

– The *user interface* that facilitates the interaction between the citizens and the Portal and acts as an entry-point to the Portal's functionalities.
– The *application layer* that implements the functionalities provided to the citizens. This layer consists of two components:
  * The Service Tree Locator (STL); and
  * The Query Mechanism.
– The *semantic repository* where all the semantic artifacts used by the Portal, e.g. the meta-ontology for STOs and the STOs, are stored. The semantic repository is based on Jena and is queried using SPARQL.

6.1. The User Interface

The user interface (UI) provides citizens with the means to interact with the Portal. Its main functionality includes presenting the questions asked by the Query Mechanism to the citizens and collecting their answers. The answers are then returned to the Query Mechanism.

More specifically, the UI implements the dynamic creation of the JSP pages that the Portal uses in order to interact with the citizens. It is important to clarify that all information that is made available through the UI, e.g. list items in dropdown lists, questions and possible answers etc., comes from the underlying STOs. Thus, the UI is able to transform the questions and their related information, which are sent to the UI by the Query Mechanism, from simple strings into html pages, so that they can be presented to the citizen.

Moreover, the summary report of the service (service description) that is presented to the citizen, after the public service is found is made available through the UI. Finally, in case the citizen decides to invoke the electronic public service the UI plays the role of the service interface, thus allowing the citizen to insert the information needed for the service execution.

6.2. The Application Layer

The Application Layer components, namely the Service Tree Locator and the Query Mechanism are discussed in this section.

The Service Tree Locator (STL) helps citizens to find the appropriate STO that models the public service which addresses their need. Citizens can use keywords to describe the service that they need. The Portal receives these keywords and sends them to the STL. Then, using these keywords the STL queries the semantic repository and finds the matching STLs. The STL contacts Wordnet in order to find synonyms and hypernyms/hyponyms for the keywords entered by the user, thus making the keyword search more effective. SPARQL queries are then formed. Finally, the relevant STLs are returned to the citizens in order for them to select the appropriate one.

The Query Mechanism (QM) is the core component of the Portal as it carries out the traversal of the STO. During the traversal of the STO, the public service that the citizen has selected is being personalized according to his/her profile. This is achieved by resolving the generic service type into the appropriate service version.

The QM traverses the STO that corresponds to the selected public service and for each loop it has to decide the next step. It is important to highlight the fact that each time the next step option is unique. This means there is no case where the same citizen could follow two different paths in the same STO.

If the current node is an *InternalNode* then the QM has to verify the conditions of all its descendants,

which are expressed as SPARQL queries. Therefore, the QM takes the appropriate question from the STO and forwards it to the UI so that the question can be displayed to the citizen.

In case the current node is a *LeafNode* (it has no descendants), then the end of the structured conversation has been reached. By now the Portal has collected all the necessary information for identifying the specific public service version that matches the citizen's profile and for deciding on his/her eligibility for it. In case the citizen is not eligible for one of the service versions that are modeled in the STO (*isNotEligible is set to true)*, then the QM terminates its execution and returns a message. The message explains to the citizen the reason why he/she is not eligible for the service, e.g. "You are not eligible for this service because you are under 18 years old".

The traversal process is summarized using pseudocode in Table 2.

A snapshot of the online structured dialogue is shown in Fig. 4. A question is asked to the citizen, in this case an entrepreneur that wants to find out if he has to register to the Chamber of Commerce and what is required for doing so. The citizen should select his profession from a drop-down, which is populated with the instances of an OWL ontology that contains different business types/activities.

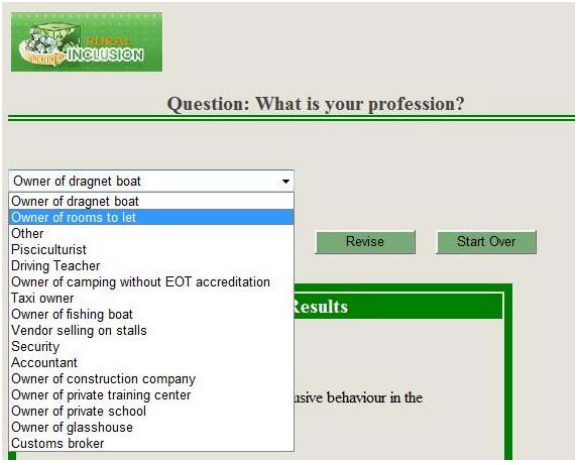

Fig. 4. Snapshot of the structured dialogue.

Table 2.

The QM's algorithm for parsing STOs.

```
BEGIN
    Let IN be the set of InternalNodes,
    Let LN be the set of LeafNodes such as IN ∪ LN = N, where
N the set of Node instances defined in the STO
    Let root be the first IN of the STO
    Let curr be the Node (either InternalNode or LeafNode) to be
processed
    Let validated be a variable that stores the result of the evalua-
tion of IN's SparqlQuery
    Let ServiceBasedUserProfile be the citizen ontology instance
that models the profile of a citizen/business
    curr := root
    while (curr ∉ LN)
            validated := false
            askQuestions(curr)
            ServiceBasedUserProfile := readAnswers()
            foreach descendant d of curr
                    if (evaluate(d)=true)
                            curr := d
                            validated := true
                            break
                    end_if
            end_foreach
            if (validated = false) //in case there is no valid descen-
    dant node
                    break
            end_if
    end_while
     showPublicServiceDescription(curr)
END
```

7. Conclusions

This paper presented an ontology-based conversational eGovernment portal, which aims at improving the informative phase of public service provision and lessening the burden of the citizens by:

- Checking the eligibility of the citizens for a specific public service before the actual execution of the service, thus saving them time, effort and money; and
- Personalizing the public service related information according to the profile and the specific needs and wants of the citizen and identify the specific public service version, thus providing targeted, tailored and comprehensive information;

The Portal's architecture is modular and as such easily extendable. It is decoupled from the service execution environment that may be available in different technologies and communicates with it using

Web Services.

The “Intel inside” the Portal is basically the different ontologies that it uses, mainly the STOs. The meta-ontology ensures reusability and extensibility as it is easy for the service providers to add new service dialogues as STOs.

Concluding, the use of OWL as the foundational technology for the development a conversational governmental portal allowed us to:

- Make use of the power provided by ontologies and reasoning. Thus the online structured dialogue does not have to comprise of predefined paths, but each time the next step is defined on the fly depending on the answers of the citizen.
- Update the Portal’s content in a plug-n-play way. Once a new STO is added, a new public service is automatically made available through the Portal.
- Reuse ontologies already available online as building blocks for the STOs, for example ontologies that model citizens or public services etc.
- Connect easily with different Web service execution environments, where the Portal provides an interface that gathers the necessary semantics from the citizens, which will then be used for the discovery and execution of Web services.
- Allow the Portal to be used with small customization in different domains beyond eGovernment. This means that once STOs that model services from other domains, e.g. eBanking, eBusiness, are developed the Portal can process them without requiring any changes in its core components, e.g. Query Mechanism.

References

[1] V. Peristeras and K. Tarabanis, The Governance Architecture Framework and Models, in: Advances in Government Enterprise Architecture, Pallab Saha, ed., IGI, 2008.

[2] FIT Project, D4: Identification of typical problems in eGovernment portals, Project Deliverable. http://www.fit-project.org/D4.pdf, 2006.

[3] J. Cardoso, The Semantic Web Vision: Where Are We?, IEEE Intelligent Systems 22(5) (2007), 84-88.

[4] N. Loutas, V. Peristeras and K. Tarabanis, Providing Public Services to Citizens at the National and Pan-European level using Semantic Web Technologies: The SemanticGov Project, in: Proc. 6th Eastern European eGov Days, 2008.

[5] N. Loutas, L. Giantsiou, V. Peristeras and K. Tarabanis, A Semantically Enabled Portal for Facilitating the Public Service Provision, in: Semantic Technologies for E-Government, T. Vitvar, V. Peristeras and K. Tarabanis eds., Springer, 2010, pp. 287-314.

[6] P. Karamolegkos, N. Loutas, N. Manouselis, E. Kalampokis, V. Peristeras and K. Tarampanis, S. Palomares Fernandes, Rural Inclusion: Alleviating Rural SMEs from administrative burdens, through sophisticated information elicitation process, in Proc. of eChallenges 2010 Conference, 2010 (to appear).

[7] X. Fang, O. L. Sheng and M. Chau, ServiceFinder: A Method Towards Enhancing Service Portals, ACM Transactions on Information Systems, 25 (4) (2007).

[8] G. M. Sacco, Interactive Exploration and Discovery of eGovernment Services, in: Proc.: 8th Annual International Digital Government Research Conference, 2007, 190-197.

[9] M. Stollberg and M. Muth, Service Customization by Variability Modeling, in: Proc. 5th International Workshop on Engineering Service-Oriented Applications (WESOA’09) co-located with the ICSOC-ServiceWave 2009.

[10] E. Tambouris, M. Vintar, and K. Tarabanis, A life-event oriented framework and platform for one-stop government: The OneStopGov project, in: Proc.: of 4th Eastern European eGov Days, 2006.

[11] K. Schmidt, L. Stojanovic, N. Stojanovic, S. Thomas, Personalization in e-Government: An Approach that combines Semantics and Web 2.0, in: Semantic Technologies for E-Government, T. Vitvar, V. Peristeras and K. Tarabanis eds., Springer, 2010, pp. 261-285.

[12] OMG, Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, OMG Available Specification, http://www.omg.org/spec/SBVR/1.0/PDF/, 2008.

[13] V. Peristeras, S. Goudos, N. Loutas, and K. Tarabanis, An ontological representation of public services: models, technologies and use cases, in: Journal of Web Engineering, 8 (3) (2009), 245-267.