CONFIGURING E-GOVERNMENT SERVICES USING ONTOLOGIES

Dimitris Apostolou[1]*, Ljiljana Stojanovic[2], Tomas Pariente Lobo[3], Jofre Casas Miró[4], Andreas Papadakis[5]

**Author to whom correspondence should be addressed; [1]PLANET S.A., Louise Riencourt 64, 11523, Greece, dapost@planet.gr, +302106905000; [2]Forschungszentrum Informatik, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany; [3]INDRA, Avda. De Brusselas, 35, 28108 Alcobendas, Spain; [4]Institut Municipal d'Informàtica, Ajuntament de Barcelona, Ajuntament de Barcelona,Av. Diagonal, 220 4a. 08018. Barcelona; [5]Archetypon S.A., 236 Sygrou Av., 176 72 Athens, Greece*

Abstract: The increasing complexity of e-Government services demands a correspondingly larger effort for management. Today, many system management tasks, such as service verification and re-configuration due to changes in the law, are often performed manually. This can be time consuming and error-prone. The main objective of the OntoGov (IST-2002-507237) project is to overcome the above mentioned problems by developing a semantically-enriched platform that will facilitate the consistent configuration and re-configuration of e-Government services. This paper outlines the overall OntoGov platform and demonstrates how the Service Modeller can be used to consistently model e-Government Services.

Key words: semantic technologies, e-Government services, ontologies.

1. INTRODUCTION

In order to fully realise the e-Government potential for productivity growth (Liikanen, 2003), it is not sufficient to modernise the front office by offering public services over the Internet through e-Government portals. Problems arise from the wide gap and inconsistencies that exist between the perspective of policy makers and public administrations' managers on the one hand and the technical realization of e-Government on the other hand. For instance, a change in policy or legislation that affects a particular Public

Administration (PA) business process does not propagate seamlessly into the corresponding e-Government service provided via the portal. Furthermore, problems arise from the loss of critical knowledge about the service configuration. Hence, for e-Government initiatives to succeed, in addition to modernising the front office by offering public services via Internet portals, attention should be also paid to streamlining, re-organising and supporting the back-office processes of public administrations that provide services to citizens. Furthermore, actions should be taken to limit the loss of critical knowledge assets during the life cycle of e-Government services.

The main objective of the OntoGov (IST-2002-507237) project is to develop a semantically-enriched platform that will facilitate the consistent configuration and re-configuration of e-Government services. This paper outlines the overall OntoGov platform and demonstrates how the Service Modeller, an intermediary project result, can be used to consistently configure an e-Government Service. The remaining of this paper is organised as follows: In Section 2, state of the art and related projects are reviewed. In Section 3, the project's software platform is outlined. In Section 4, an overview of the service modelling approach is presented while in Section 5, service configuration is illustrated using a real-life scenario. Finally, the benefits of our approach are outlined in Section 6.

2. STATE OF THE ART AND RELATED PROJECTS

State of the art in e-Government includes realising the concept of one-stop e-Government (Hagen and Kubicek, 2000), especially together with the idea of service portals with life-situation navigation (Tambouris and Wimmer, 2004). The basic ideas of one-stop e-Government are already well-developed and their technical realisation on top of state-of-the-art Web Service technology. What is not solved sufficiently, are the methodological and technological prerequisites as well as the back-office processes, which help turning one-shot investments into one-stop approaches into sustainable, long-term endeavours which can be maintained effectively and consistently over a longer period of time. This idea requires on one hand a higher level of re-configurability and on-the-fly changes of services – which is not provided by today's web service technology; and on the other hand a well-understood and technically supported knowledge logistics along the horizontal dimension (many implementing sites) and vertical dimension (several levels of decision-making).

To deal on the one hand with re-configurability and changes of e-Government services and on the other hand with knowledge-enhanced back-office processes for configuring eGov services, we need tools based on robust conceptual models. In OntoGov, we are using Semantic Web technologies for constructing ontologies, which represent the meaning of processed data and resources and provided functionality of e-Government services. Ontologies have been employed by other projects in the e-Government domain, each with a primary objective: The e-POWER project (van Engers et al., 2002) has employed knowledge modelling techniques for e.g. consistency checks, harmonisation or consistency enforcement in legislation. The SmartGov project (Adams et al., 2003) developed a knowledge-based platform for assisting public sector employees to generate online transaction services by simplifying their integration with already installed IT systems. Similarly, the ICTE-PAN project (Loukis et al., 2003) developed a methodology for modelling PA operations, and tools to transform these models into design specifications for an e-Government system. Further, there are a number of ongoing projects e.g. Terregov (Benamou, 2004), Qualeg (Tatsiopoulos, 2004), that make use of semantic technologies for achieving semantic interoperability and integration between e-Government systems.

3. ONTOGOV PLATFORM OVERVIEW

In principal, the lifecycle of an e-Government service starts when PA Managers trigger the generation or the change of a service. In order to accomplish this task, PA Managers need to have a high-level view of the service model, links to related laws, resources involved and inter-relations with other services. Such a high-level view is provided by the service models developed through OntoGov's Ontology Management System (see Figure 1).

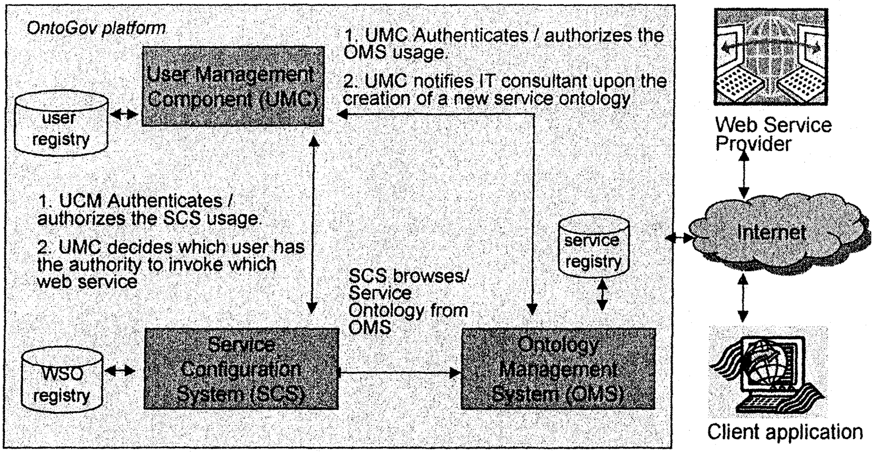
OntoGov platform
User Management Component (UMC)
user registry
1. UMC Authenticates / authorizes the OMS usage.
2. UMC notifies IT consultant upon the creation of a new service ontology
1. UCM Authenticates / authorizes the SCS usage.
2. UMC decides which user has the authority to invoke which web service
SCS browses/ Service Ontology from OMS
service registry
WSO registry
Service Configuration System (SCS)
Ontology Management System (OMS)
Web Service Provider
Internet
Client application

Figure 1. Bird's eye view of the OntoGov platform

The **Ontology Management System (OMS)** is used for creating, modifying, querying, and storing ontology-based descriptions of e-Government services. It focuses on the service lifecycle management, which includes service modelling, service reuse, service discovery, service composition as well as service reconfiguration. The OMS comprises: (i) The ***Service Modeller***, an editor for the semantic description of e-Government services. It has a graphical user interface that enables domain experts to create and maintain service ontologies. (ii) The ***KAON2 OIModeller***, a graphical tool for general-purpose ontology creation and maintenance. (iii) The ***Service Registry*** that enables the registration and searching for ontologies.(iv) The ***Service API*** that provides capabilities for the automatic identification of problems (i.e. inconsistencies) in the description of the e-Government services that can arise during the modelling or changes in relevant data (e.g. in the law). When such problems arise, it assists the domain experts in identifying the sources of the problem, in analysing and defining solutions for resolving them. Finally, it helps in determining the ways for applying the proposed solutions.

The role of the **Service Configuration System (SCS)** is to bridge the gap between the service definition provided by the OMS, and the generation, deployment and execution of the e-Government service. Configuration, generation and deployment of the e-Government service is handled by the SCS Configuration Framework (Figure 2), while execution of services is handled by the SCS Runtime Framework (Figure 3).

The result of the OMS is a set of ontologies that defines among other things the service model and the domain information needed to provide a service. The service model is defined through the process ontology and becomes the main source of information for the Configuration Framework. This information is transformed to a machine-readable service description allowing its execution on a server. This task is performed through the ***SCS Builder*** component.

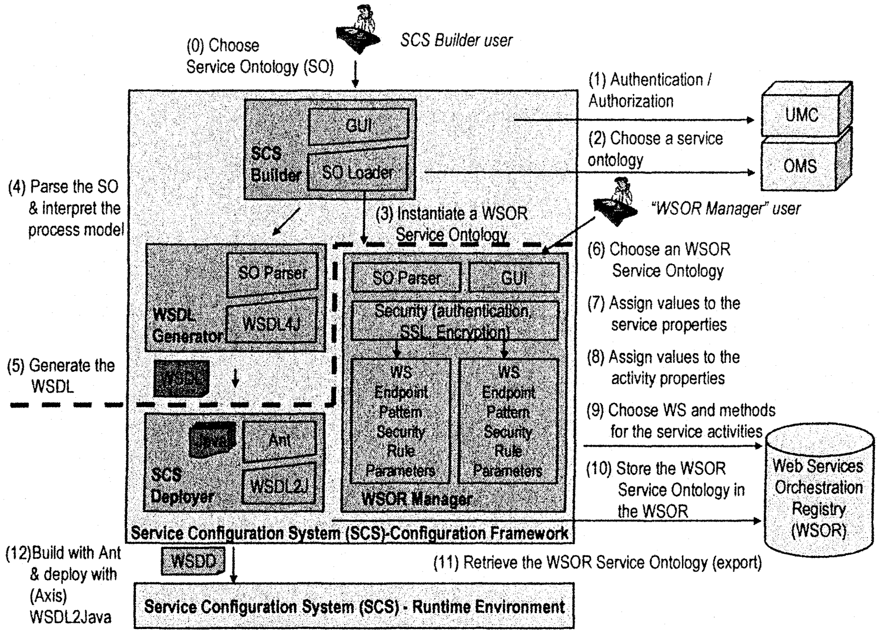
(0) Choose
Service Ontology (SO)
SCS Builder user
(1) Authentication / Authorization
(2) Choose a service ontology
UMC
OMS
"WSOR Manager" user
GUI
SCS Builder
SO Loader
(4) Parse the SO & interpret the process model
(3) Instantiate a WSOR Service Ontology
SO Parser
WSDL Generator
WSDL4J
SO Parser
GUI
Security (authentication, SSL Encryption)
WS Endpoint Pattern Security Rule Parameters
WS Endpoint Pattern Security Rule Parameters
WSOR Manager
(5) Generate the WSDL
WSDL
Java
Ant
SCS Deployer
WSDL2J
(6) Choose an WSOR Service Ontology
(7) Assign values to the service properties
(8) Assign values to the activity properties
(9) Choose WS and methods for the service activities
(10) Store the WSOR Service Ontology in the WSOR
Web Services Orchestration Registry (WSOR)
Service Configuration System (SCS)-Configuration Framework
(12)Build with Ant & deploy with (Axis) WSDL2Java
WSDD
(11) Retrieve the WSOR Service Ontology (export)
Service Configuration System (SCS) - Runtime Environment

Figure 2. Functional Architecture of the Service Configuration Framework – Configuration Framework

The ***Web Service Orchestration Registry (WSOR)*** is an ontology-based repository where the mapping among activities of the service model and their implementation (Web services) is performed.

The SCS Runtime Framework, allows the execution of the service. Within the SCS Runtime Framework the end point derives the request to the ***Process Engine*** component, which is a workflow tool that is in charge of querying the Service Ontology (process model information) and selecting the first activity described in the process model. The Process Engine sends control to the ***WS Manager***. The WS Manager looks up in the WSOR ontology the implementation that, according to the data provided by the consumer or derived from the process, should be invoked to accomplish the activity selected. The ***Synchronization Manager*** hides the complexity of dealing with synchronous/asynchronous calls. Execution and monitoring (logging) of the service is done by the ***SCS Audit and Tracking component***.

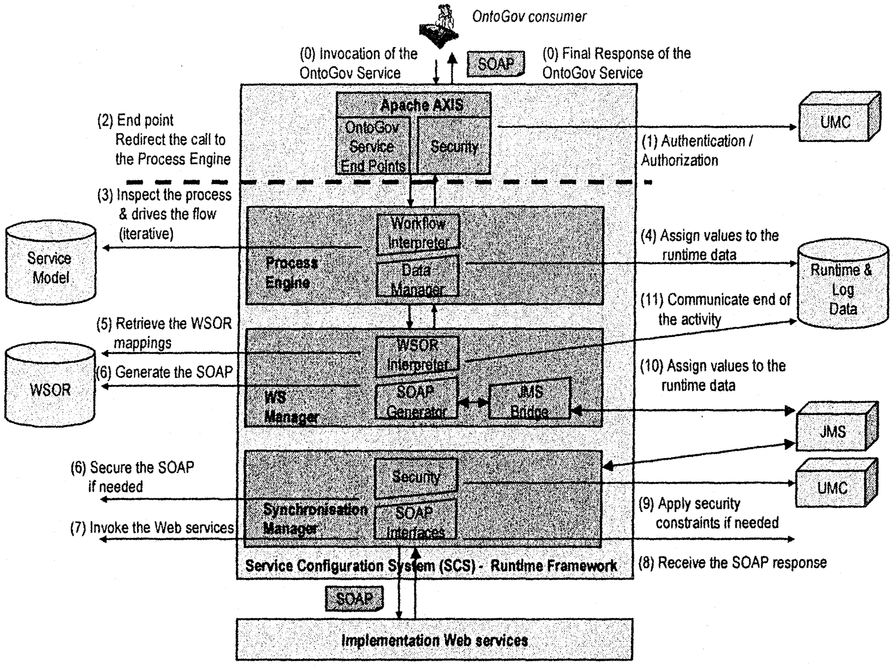
OntoGov consumer
(0) Invocation of the OntoGov Service
SOAP
(0) Final Response of the OntoGov Service
Apache AXIS
OntoGov Service End Points
Security
UMC
(1) Authentication / Authorization
(2) End point Redirect the call to the Process Engine
(3) Inspect the process & drives the flow (iterative)
Service Model
Workflow Interpreter
Process Engine
Data Manager
(4) Assign values to the runtime data
Runtime & Log Data
(11) Communicate end of the activity
(5) Retrieve the WSOR mappings
WSOR
(6) Generate the SOAP
WSOR Interpreter
SOAP Generator
JMS Bridge
WS Manager
(10) Assign values to the runtime data
JMS
(6) Secure the SOAP if needed
Security
UMC
Synchronisation Manager
SOAP Interfaces
(9) Apply security constraints if needed
(7) Invoke the Web services
Service Configuration System (SCS) - Runtime Framework
(8) Receive the SOAP response
SOAP
Implementation Web services

Figure 3. Functional Architecture of the Service Configuration Framework – Runtime Framework

The acceptance of the OntoGov approach poses strict security and management requirements. In order to leverage the OntoGov functionality the PAs are expected to expose some of their resources and to allow access to their procedures in an IT environment. Such an open and flexible approach can only be built on a reliable platform of trusted relationships. The **User Management Component (UMC)** supports the areas of User and Group management, Authentication, Access control and Messaging – extending over the security considerations. It is an auxiliary component of the platform, supporting a personalized working environment, the co-existence of the components and the shielding of sensitive resources.

4. SERVICE MODELLING IN ONTOGOV

Based on the analysis of the existing standard for Semantic Web Services (i.e. OWL-S and WSMO) and by taking into account the e-Government specific requirements (Stojanovic et al., 2004), we have defined a ***meta ontology cluster*** that contains general ontologies that may be used for describing e-Government services and do not change from one deployment to another. It consists of following ontologies:

- the ***Legal Ontology*** defines the structure of the legal documents, which includes paragraphs, sections, amendments, etc.;
- the ***Organisational Ontology*** models an organisation by defining its organisational units, roles, persons, resources etc.
- the ***Lifecycle Ontology*** comprising instances of all (design) decisions relevant for the new service (e.g. technical or process immanent reasons), including instances of the legal and organizational ontologies;
- the ***Domain Ontology*** contains domain specific knowledge;
- the ***Service Ontology*** describes the elements for modelling the service flow. It includes the *Domain Ontology* for defining inputs and outputs as well as the *Lifecycle Ontology* for explaining reasons that motivate the decisions;
- the ***LifeEvent Ontology*** models the categorization of the e-Government services;
- the ***Profile Ontology*** contains metadata about e-Government services and includes all previously mentioned ontologies.

The *Profile Ontology* and the *Service Ontology* are defined based on the corresponding OWL-S ontologies by taking into account the e-Government specificities such as a reference to the law that is modelled through the *Legal Ontology*. The *Domain Ontology* defines the "terminology" used in the e-Government domain (e.g. type of documents such as passport). The *Organisation Ontology* is defined to take into account experiences from the business process modelling and reengineering, since changes in the organizational structure can cause changes in the process model. The *LifeEvent Ontology* is specific for the e-Government domain and it is defined to support better searching for E-Government services.

The *Lifecycle Ontology* is defined to help the domain expert introduce the changes in the service description and to document the reasons for these changes (Apostolou et al. 2005). This means that in the OntoGov project we use ontologies not only for describing and composing services provided by public administrations, but also for modelling dependencies between decisions of the different stakeholders (e.g. politicians, public managers and software developers) in order to make services easier to develop and maintain.

5. ILLUSTRATION OF THE USE OF THE ONTOLOGY MANAGEMENT SYSTEM

In this section we illustrate how an e-Government service is modelled using the Ontology Management System on the basis of an example. In the example we use the real service "Minor Building Work License" as this is being offered by the municipality of Barcelona. In the past, there were two kind of licenses; the citizen had to bring the documentation to the Citizens Attention Office and the municipal architects had to study it. It could take from 1 to 4 months, due to the large amount of licenses requested. Currently, the municipality of Barcelona is involved in an e-Government process reengineering to simplify the service. In order to achieve this goal, two main steps are followed:

1. Change in the legal municipal normative in order to simplify the procedure. Moreover, a new type of building licence is being introduced.
2. Use of OntoGov platform and other tools so that time and human action in the procedure is reduced as much as possible.

Essentially the new service will be able to detect the kind of building construction the citizen wants to perform and act accordingly. For instance, if a citizen wants to change the structure of a building, s/he will be asked to provide documents related to the security of the works, a project signed by an architect, etc. However, if s/he only wants to change colour of the façade, he may get the license immediately. But what happens if the citizen wants to change the façade's colour of the building placed in the 401st of Mallorca Street? The service will be able to notice the difference: in that number there is the Sagrada Família temple. Therefore, there are many variables to take into account when a citizen requests a minor building construction. This automation is provided in a number of service steps as these are described in Table 1.

Table 1. "Minor Building Work License" service steps

No	Service Activity
1	Ask the citizen about the address and the kind of the works
2	Decide if they are major or minor works (the service is just offered for the minor works case)
3	In case of being minor works, perform more concrete questions about the works
4	Decide which one of the 3 different licenses includes this kind of minor works
5a	If it is an "Assabentat" (informed regime): - store the data related to the works - grant the license to the citizen.
5b	If it is a "Comunicat" (prior notification regime): - store the data related to the works - store the citizen data

No	Service Activity
	- display the list of documents the citizen must provide before receiving the license. - display the payments he must perform before receiving the license.
5c	If it is a "Llicència" (standard license): - store the data related to the works - store the citizen data - display the list of documents the citizen must provide - warn the municipal architects about the procedure so that they can study it in depth.

The development of this service with ONTOGOV's OMS will be as follows:

The domain expert extends and creates instances of the meta-ontologies for the following ontologies:

- Domain ontology, comprising concepts like data (e.g. kind of highway, name of the highway, number, boolean answers to questions related to the works, etc.) and documents (e.g. application form, informed regime license, etc.)
- Legal ontology, comprising instances of process relevant law or regulations, e.g. basis of this service is the new municipality's ordinance governing minor works (1-1-2005). Then several instances will be initiated in the legal ontology indicating the related law[1] ('Ordenança reguladora d'obres menors'), the title ('Títol 2: Règim d'assabentat') and article ('Article 13: Abast i presentació del règim d'assabentat').
- Organisational ontology, comprising instances of process relevant organizational units, e.g. involved in the service are the organizational units 'I.M.I.', 'Municipality of Barcelona', and 'Town-planning department' with its roles and personnel.

[1] Note: example is taken from the Catalan legislation

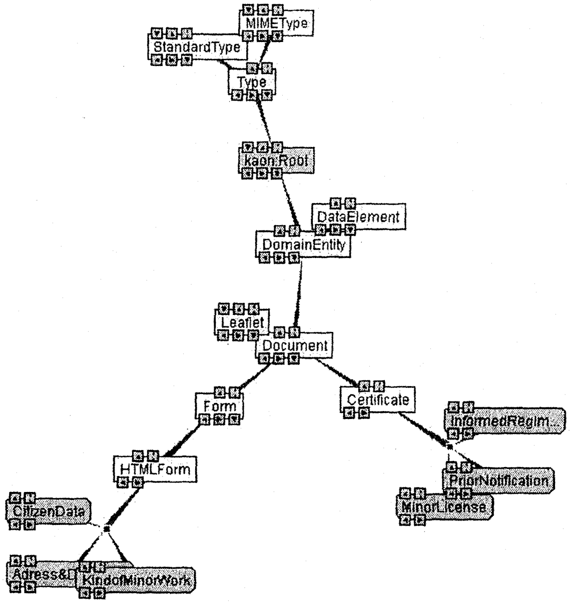

Figure 4. Parts of the Domain Ontology

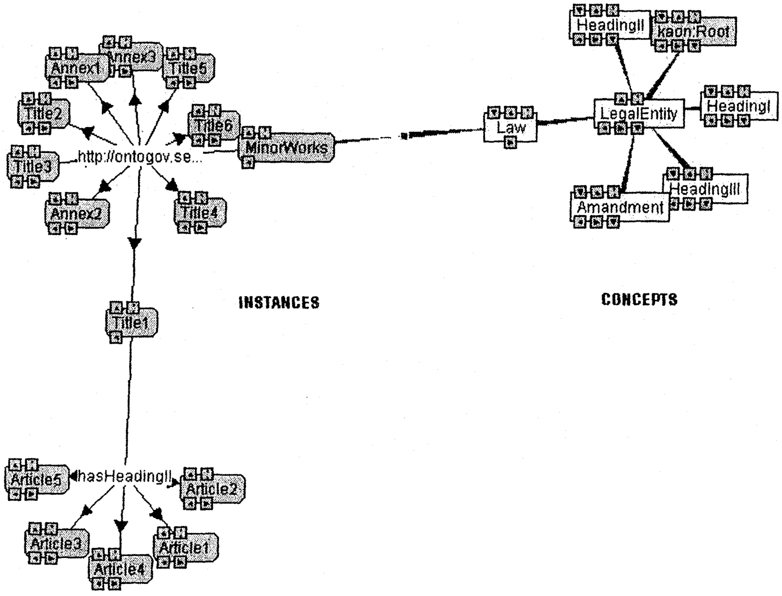

Figure 5. Parts of the Legal Ontology

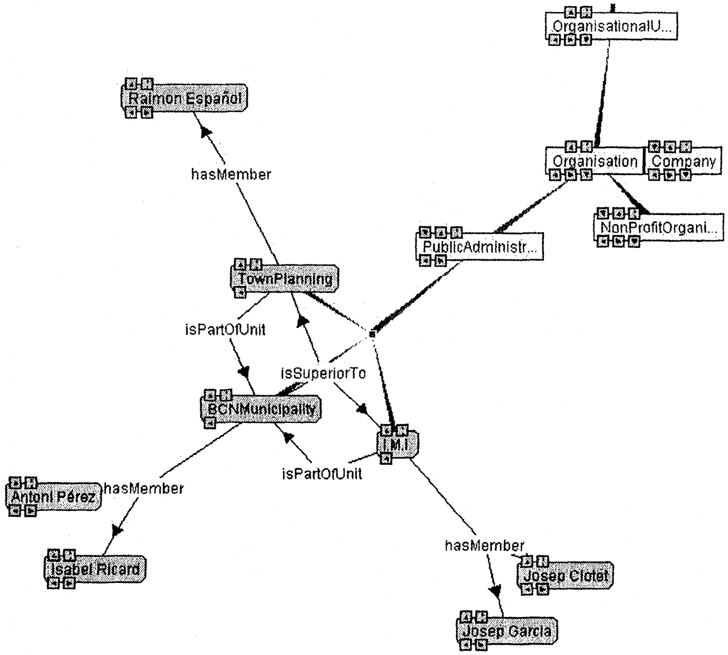

Figure 6. Parts of the Organisational Ontology

Furthermore, the Domain Expert designs the service using the Service Modeller. First of all s/he creates the service model by naming it and including the ontologies related to it (already modelled with the OI-Modeller from the meta-ontologies). Then s/he must think about the process workflow and perform four main actions during the modelling process:

- To decide which atomic services he will need and place them into the graph.
- To establish the relationships between these atomic services by using the different options available in the OntoGov's Service Modeller (split, join, switch, etc.).
- To keep the consistency between the outputs and the inputs of every atomic service and its following ones.
- To establish at least one first service and one last service so that the workflow may be understandable by the Ontogov's Service Modeller.

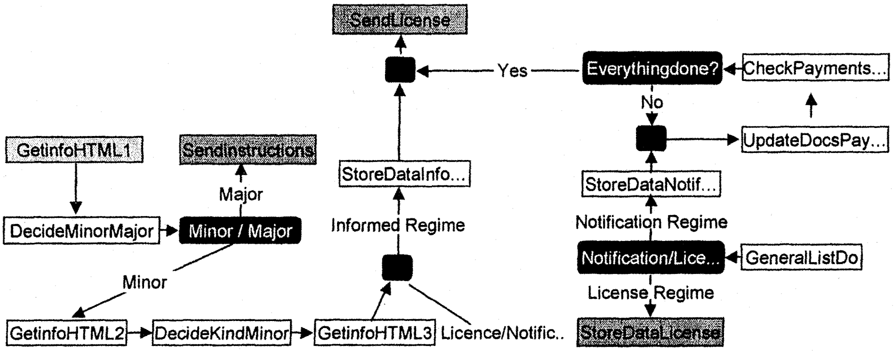

Figure 7. "Minor Building Work License" service

After the design process is completed, it is the task of the Service Configuration System to generate the definition of the new service while the User Management Component will notify the IT consultant that a service model has been created and is ready to be implemented and deployed (not described herein).

6. CONCLUSIONS

In this article, we highlighted a novel application of semantic technologies in the e-Government domain: utilising semantics to support the consistent configuration and change management of e-Government services. The benefits of utilising ontologies in the configuration of e-Government services include the ability to perform consistency checks. The following consistency checks are currently being performed by the Service Modeller:

(C1) Each service has to have a reference to at least one business rule (law). E.g. The Minor Building Work License service has reference to the municipality's ordinance governing minor works (1-1-2005). C1 enables to find the corresponding service if a law is changed

(C2) Each service has to have at least one resource that controls its execution. E.g. The Minor Building Work License service has reference to the 'Town-planning department' with its roles and personnel.

(C3) Each service has to have at least one software component attached to it that implements it. E.g. The Minor Building Work License service has reference to the 'Building Database' (containing information about city buildings as well as list of the documents the citizen must provide for every kind of construction works) and to the HOST database (containing the current state of the payments).

(C4) Each service has to have at least one input. E.g. the kind of construction works to be performed.

(C5) Each service has to have at least one output. E.g. which of the 3 different types of licenses apply (if work is classified under 'minor works').

(C6) Each service input has to be either output of some other service or is specified by the end-user. C6 ensures that a change in an output of an activity is propagated to the inputs of successor activities and vice versa

(C7) If the input of a service is the output of another service, then it has to be subsumed by this output.

(C8) If the input of a service subsumes the input of the next service, then its preconditions have to subsume the preconditions of the next one. C8 prohibits the changes which lead to non-optimal service reconfiguration. For example, if the preconditions for 'Decide Kind Minor' include a constraint that the building has not to be preservable, the preconditions of the next activity cannot be that the building is historical.

(C9) If two services are subsumed by the same service, then their preconditions have to be disjoint.

(C10) If a service specialises another service, one of its parameters (i.e. inputs, outputs, pre- or post-conditions) has to be different. The difference can be achieved either through the subsumption relation with the corresponding counterpart or by introducing a new one.

(C11) Inputs, outputs, pre- and post-conditions have to be from the domain ontology.

(C12) Any specialization of the activity A1 must always be a predecessor of any specialization of the activity A2, where A1 and A2 are two activities defined in the Meta Ontology and their order is given in advance (i.e. A1 precedes A2).

Further benefits of our semantics-based approach include the support for conflict resolution and change propagation. Based on the consistency checks described above, the system, by providing an inconsistencies discovery mechanism, will notify the domain experts about logical conflicts. Moreover, it will provide enough information to analyse the sources of conflicts. Its role will be to inform a domain expert about the necessity for updating an e-Government service, and to allow the application of the service changes, enabling an easy spotting of potential problems. Finally, the system will propagate changes from the changes in business rules (e.g. laws) to the changes in the semantic web services and within services. This will guarantee the transfer of all dependent service ontologies into another consistent state.

Finally, in case of a change of a law the OntoGov system can be queried to retrieve affected service activities. In our example, "GenerateListDocs" (displays the list of documents the citizen should have provided before receiving the license) and "DecideKindMinor" (decides which kind of license must be granted) are based on various decisions. The different documents to be provided before starting the works are defined by law, depending on each kind of works. The heritage catalogue level of a certain building or the affectations of the street where this building is placed (they are both necessary in order to decide the kind of the license) are located in the town-planning department databases, whereas the citizen data and the state of the payments must be stored in the Municipality HOST. If a new government is stricter in the security within the building works and modifies the existing law affecting it, the list of documents the citizen must provide will change. The OntoGov system will search for all service implementation decisions based on this legal reason and, as a result, all affected services and activities will be listed and proposed for modification. In the example, this is the service "Minor Works License" with its activity "GenerateListDocs". Similarly, if some building loses its heritage catalogue level, the activity "DecideKindMinor" will be affected.

ACKNOWLEDGEMENT

The research presented in this paper was financed by the EC in the project "IST PROJECT 507237 - OntoGov".

REFERENCES

Adams, N., S. Haston, A. Macintosh, J. Fraser, A. McKay-Hubbard, and A. Unsworth, (2003) SmartGov: A Knowledge-Based Design Approach to Online Social Service Creation, in Bramer, M., Ellis, R., Macintosh, A; (eds.). 'Applications and Innovations in Knowledge-Based Systems and Applied Artificial Intelligence XI'; Proceedings of AI-2003 the 23rd Annual International Conference of the British Computer Society's Specialist Group on Artificial Intelligence ; Peterhouse College, Cambridge, UK, 16th-17th December, 2003.

Apostolou, D., L. Stojanovic, T. Pariente Lobo, B. Thoenssen, (2005) Towards a Semantically-Driven Software Engineering Environment for eGovernment, TCGOV 2005, pp. 157-168, 2005.

Benamou, N. (2004) "Terregov project overview", IANOS conference, Budapest, 2004.

Hagen M., Kubicek H. (2000) One-Stop-Government in Europe: Results of 11 national surveys, University of Bremen, Bremen, Available at http://infosoc2.informatik.uni-bremen.de/egovernment/cost/one-stop-government/home.html, 2000.

Liikanen, E., (2003) e-Government and the European Union, speech by Mr Erkki Liikanen (Member of the European Commission, responsible for Enterprise and the Information Society) at the Internet and the City Conference "Local eGovernment in the Information Society" Barcelona - 21 March 2003.

Loukis, E., S. Kokolakis, (2003) Computer supported collaboration in the Public Sector: the ICTE-PAN Project, in preceedings of eGOV 2003 / DEXA 2003.

Stojanovic, L. et al.,(2004a) available as Deliverable D2, http://www.ontogov.com, 2004.

Tambouris E., Wimmer M. (2004) Online one-stop government: a single point of access to public, in *Digital Government: Strategies and Implementations in Developed and Developing Countries*, Wayne Huang, ed., Idea Publishing Group, 2004.

Tatsiopoulos, C., (2004) "QUALEG approach to intra-government interoperability", in Workshop on Technological and architectural challenges, eGOV04 Conference, Zaragoza Spain, 1st September 2004.

van Engers, T., J. M. Patries, J. Kordelaar, J. den Hartog, E. Glasséee, (2002). Available at http://lri.jur.uva.nl/~epower/ (March 2005)