Andreas Abecker
Gregoris Mentzas
Ljiljana Stojanovic
(Editors)

Proceeding of the Workshop on

Semantic Web for eGovernment 2006

Workshop at the 3rd European Semantic Web Conference

12 June 2006, Budva, Serbia & Montenegro

Content

A Semantic Web GIS based Emergency Management System*

Vlad Tanasescu[1], Alessio Gugliotta[1], John Domingue[1],
Leticia Gutierrez Villarias[2], Rob Davies[2], Mary Rowlatt[2], Marc Richardson[3]

[1] Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA, UK
{v.tanasescu, a.gugliotta, j.b.domingue}@open.ac.uk

[2] Essex County Council, County Hall,
Chelmsford, CM1 1LX, UK
{Leticia.gutierrez, maryr}@essexcc.gov.uk
rob.davies@mdrpartners.com

[3] BT Exact
Adastral Park Martlesham, Ipswich IP5 3RE, UK
marc.richardson@bt.com

Abstract. Public organisations access spatial-related data for management as well as for communication purposes. The approach of using traditional Geographical Information Systems (GIS) is not always satisfactory; users have to cope with distributed heterogeneous data sources to find appropriate resources for particular situations. Developments in the field of Semantic Web Services (SWS) show the opportunity of adding higher semantic levels to the existing frameworks, to improve their usage and ease scalability. We outline a Semantic Web GIS in which data sources and services are made available through SWS, described by ontologies, allowing interoperability as well as reasoning to create a comprehensive response adapted to user goals. The Emergency Management System described in this paper as a practical example of Semantic Web GIS instantiation provides a goal oriented tool for emergency planners.

1 Introduction

In an emergency situation, multiple agencies need to collaborate, sharing data and information about actions to be performed. However, many emergency relevant resources are not available on the network and interactions among agencies or emergency corps usually occur on a personal/phone/fax basis. The resulting

* This work was supported by the DIP (Data, Information and Process Integration with Semantic Web Services) project. DIP (FP6 - 507483) is an Integrated Project funded under the European Unions IST programme.

interaction is therefore limited in scope and slower in response time, contrary to the nature of the need for information access in an emergency situation.

Emergency relevant data is often spatial-related, and *Spatial-Related Data (SRD)* is traditionally managed with the help of *Geographical Information Systems (GIS)*. GIS allow access to different layers of SRD such as highways, transportation, postal addresses index, land use, etc. GIS support decision making by facilitating the integration, storage, querying, analysis, modeling, reporting, and mapping of this data.

Unfortunately, GIS are often centralized and isolated systems, and heterogeneity arises in the way different organisation collect and manage data, according to a particular view of the world. This is often a barrier to SRD exchange. The lack, and maybe the impossibility, of consensus about the spatial domain limits communication and knowledge of available information, leading to inaccuracies whilst introducing an increased amount of manual work. These inefficiencies can lead to disastrous consequences in an emergency situation.

To alleviate this, service-oriented architectures are becoming popular in the implementation of e-government programmes; combined to recent developments in the area of *Web Services (WS)* and the *Semantic Web* [3] they can enable the creation of agile networks of collaborating applications distributed within and across public organization boundaries ([4], [9]).

Using WS, SRD can be shared on the internet via services which become autonomous and platform-independent computational elements. Unfortunately, despite the acceptance of standards for WS description (WSDL[1]) and publishing (UDDI[2]), syntactic definitions do not completely describe the capability of a service and cannot be understood by software programs; a human developer is always required to interpret the meaning of inputs, outputs, applicable constraints as well as the context in which services can be used.

The Semantic Web aims to allow the development of easy to use applications and transparent access to services and data, by giving machine understandable meaning (semantics) to services as well as contents on the Web, and to create a universal medium for information exchange. In particular, the *Semantic Web Services (SWS)* technology provides an infrastructure in which new services can be added, discovered and composed continually [4], by combining the flexibility, reusability, and universal access that typically characterize a WS, with the expressivity of semantic markup and reasoning. This allows the invocation, composition, mediation, and execution of complex services with multiple paths of execution, and levels of process nesting.

In this paper, we describe results in the development of a Semantic Web GIS *Emergency Management System (EMS)* relying on SWS technologies. The EMS assists the *Emergency Planning Officer (EPO)* in the task of retrieving, displaying, and interacting with emergency relevant information. This can include, according to the kind of emergency, weather forecasts, available rescue corps, evacuation procedures, supplies providers, available rest centres, categories of affected and vulnerable people, nature and location of damaged or endangered facilities, access of critical spots, etc. As a result, involved agencies become able extend their knowledge about the emergency situation by making use of different functionalities based on data

[1] http://www.w3.org/TR/wsdl
[2] http://www.uddi.org/

held by other agencies which otherwise might not be accessible to them or slow to obtain manually.

Sections are arranged as follows. Section 2 gives an overview of the Semantic Web GIS framework we propose. Section 3 provides a detailed overview of the architecture of the EMS prototype. Section 4 briefly outlines its practical usage. Finally, Section 5 concludes and discusses future work.

2 A Semantic Web GIS Framework

Any information system can gain advantage from the use of semantics [10]. In GIS, the use of semantic layers, although not yet firmly established, is being investigated in a number of research studies [5], [7], [11]. Having ontologies describing the SRD repository and its functionalities is believed to make cooperation with other systems easier and to better match user needs. In order to ease the transition to a *Semantic Web GIS (SWGIS)*, we adopt WSMO[3] – a promising SWS framework – and IRS-III – a tested implementation of this standard [1] – in order to expose data sources. In the following, we briefly describe these two technologies as well as give a whirlwind introduction to GIS technologies, before describing the framework we propose.

2.1 Semantic Web Services with WSMO and IRS-III

The *Web Service Modelling Ontology (WSMO)* is a formal ontology for describing the various aspects of services in order to enable the automation of WS discovery, composition, mediation and invocation. The meta-model of WSMO defines four top level elements: *Ontologies*, *Goals*, *Web Services*, and *Mediators*.

Ontologies [2] provide the foundation for describing domains semantically. They are used by the three other WSMO components.

Goals define the tasks that a service requester expects WS to fulfil. In this sense they tend to reflect the service user's intent.

Web Service descriptions, in terms of *capabilities* (what the service can do) and *interface* (how to use it), represent the behaviour of a deployed Web Service. The description also indicates how WS communicate (*choreography*) and how they are composed (*orchestration*).

Mediators handle issues of data and process interoperability that arise between heterogeneous systems. One of the characterizing features of WSMO is that all components – Ontologies, Goals and Web Services – are linked by Mediators. In particular, WSMO provides four kinds of mediators:

- *oo-mediators* for mediating between heterogeneous ontologies;
- *ww-mediators* connect WS to WS;
- *wg-mediators* connect WS with Goals;
- *gg-mediators* link different Goals, solving input conflicts and transforming processes.

[3] http://www.wsmo.org/2004/d2/v1.0/

The incorporation of four classes of mediators in WSMO facilitates the clean separation of different mapping mechanisms.

IRS-III, the *Internet Reasoning Service* [1], is a platform which allows the description, publication and execution of Semantic Web Services, according to the WSMO conceptual model.

Based on a distributed architecture communicating via XML/SOAP messages, it provides an execution environment for SWS; ontologies are stored by the server, and used in WSMO descriptions to support discovery, composition, invocation and orchestration of WS. It allows *one-click publishing* of "standard" program code to WS by automatically generating an appropriate wrapper. Standard WS or REST services can also be trivially integrated and described by using the platform.

Also, by extending WSMO's goal and Web Service concepts, clients of IRS-III can invoke web services via goals. That is, IRS-III supports so called *capability-*, or *goal-driven* service invocation which allows the user to use only generic inputs, hiding the possible complexity of a chain of heterogeneous WS invocations.

2.2 Geographical Information Systems

A GIS allows the creation and management of objects composed of spatial attributes (polygons, nodes, maps, etc) as well as descriptive ones (names, numeric values, etc). GIS are "smart map" tools that allow users to express complex queries, visualize and analyze spatial information, as well as edit it.

Maps available on the web, for spotting an address or getting transportation information, are popular but allow only simple queries.

However, recently, a new type of mapping systems emerged; highly responsive mapping frameworks providing API (Google[4], Yahoo[5], Mapquest[6], etc.). They are also usually enhanced with "reality effects" – e.g. seamless transition between maps, satellite and hybrid views, 2.5-3D visualisations, street level photography, etc. – which make them even more appealing. API allow developers to populate online maps with custom information – location of "events" or "things" –, by collecting data from standard documents such as RDF files, or simply by ad hoc "web scraping" of HTML resources. These embryonic but very agile *Web GIS*, called *mashups*, can merge more than one data sources and add functionality such as filtering and search features.

However, although extremely popular[7], relatively easy to build and to enhance, Web GIS do not avoid traditional issues attached to non semantic applications; indeed (i) handling data heterogeneity still requires considerable manual work, (ii) the lack of semantics limits the precision of queries, and (iii) limited expressiveness usually drastically limits functionality [13].

---

4 http://www.google.com/apis/maps/
5 http://developer.yahoo.com/maps/
6 http://www.mapquest.com/openapi/
7 cf. http://googlemapsmania.blogspot.com/ for a number of mashups using Google Maps API.

2.3 A Semantic Web GIS Proposal

A *Semantic Web GIS (SWGIS)*, as sketched in [13], is a system which answers geographically oriented queries in a smart way while integrating multiple and heterogeneous information sources. As such, it needs to address the previous issues. In order to achieve this, a multi-layered architecture is needed (Fig. 1). The elements of this architecture will be discussed in turn.

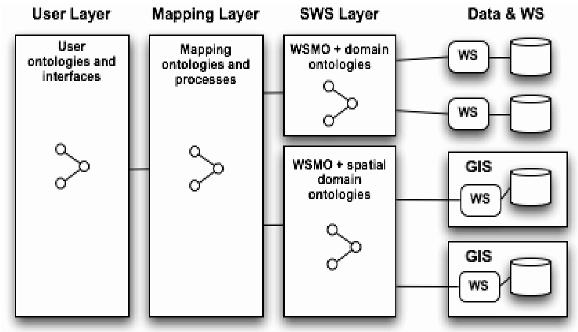

Fig. 1. A schema of the proposed Semantic Web GIS framework.

Data and Web Services Layer. The WS layer allows distributed datasets to be accessed through the network. They also hide the underlying relational access interface to provide simpler but well defined query operations.

Semantic Web Services Layer. The operations provided by the WS layer can be semantically described using the WSMO framework. However the spatial domain ontology to use is far from generating consensus. Several ontology modelling solutions have been explored in the literature [15]; typically, top-level spatial ontologies include models of topology and mereology (the formalization of *part-of* relationships), but point based set theory can also be used as a fundamental structure [16], moreover the ontological status of typical geographic objects such as *regions*, *boundaries*, *processes* or *events* as opposed to more common entities is unclear ([8], [5]). The debate also involves the importance of *graduation* and *fields* for the representation of spatial concepts, as well as describes a hiatus between the scientific notion of space and its cognitive apprehension, mostly qualitative, as studied in the field of “Naïve geography” [6].

User Ontology Layer. Although conventional GIS exhibit various levels of graphical user interface complexity, as well as custom query languages, accessing the underlying data level is often the only way to express complex needs, i.e. by using a database query language. Indeed, GIS usage can hardly be called intuitive and often requires from the user technical or even programming skills.

However, complexity has to be avoided in a semantic web context [3]. To allow this without loosing expressivity goal orientation should be combined with context in order to allow the user to access relevant goals at any moment. In SWGIS, we believe that attaching goals to objects, as described in an ontology, and using the sequence of goal invocation as well as the location of the query as a context may help simplifying the task of query specification.

Moreover, to efficiently support an activity such as emergency planning, precision is essential; only goals and data related to the emergency have to be displayed. Therefore an appropriate user ontology must capture the decision making process in terms of goals and relevant information.

Also, as the number of information sources increases, generic cognitive concepts have to be used since the user may not know beforehand what domain concept he is asking for. For example a request for “shelters” will only include heated accommodations in a snow storm context, but will have a different extension elsewhere.

From a user interface point of view, the SWGIS client may only know how to represent data up to a certain degree of specificity. To this purpose we are using a generic (archetypal) ontology including “image schematic” concepts, i.e. notions supposedly immediately understandable such as *container* or *link* [14].

In the future, qualitative reasoning features characteristic of “naïve geography” will also find room in the user ontology layer.

Mediation Layer. The mediation layer must allow a smooth and non destructive transition from the SRD and service description ontologies toward user oriented cognitively sound information, such as goals and schemas. For this purpose, this layer needs to transform some concepts but also offer an environment in which other concepts can merge, through inheritance or other multi-representation techniques, to allow multiple views of a same domain, and give the possibility of representing the same element differently according to semantic context such as task at hand, or spatial context, scale, or dimensionality (2-3D).

3 Description of the Prototype

The prototype was designed for the *Essex County Council (ECC) Emergency Planning Department*. The ECC is a large local authority in South-East England (UK). Following several interviews with SRD holders in the ECC it was decided to focus the scenario on the ECC Emergency Planning department, and more concretely, on an previous emergency situation: the snowstorm which occurred in the vicinity of Stansted airport on the 31st of January 2003. In order to avoid all interferences, data from the ECC Emergency Department and the Meteorological Office was replicated. This will also allow us to compare EPO’ decisions regarding contact with rescue corps and voluntary associations, or actions necessary to provide refuge and supplies to trapped travelers, etc. – with those of the prototype users.

The EMS prototype is a decision support system, which assists the end user – currently the *Emergency Planning Officer (EPO)*, but we believe our design extensible to other emergency corps such as ambulance service, fire service, police, etc. – in gathering information related to a certain type of event, faster and with increased precision.

3.1 Architecture

Based on the SWGIS generic framework introduced in Section 2.4, we developed the following prototype architecture (Fig. 2).

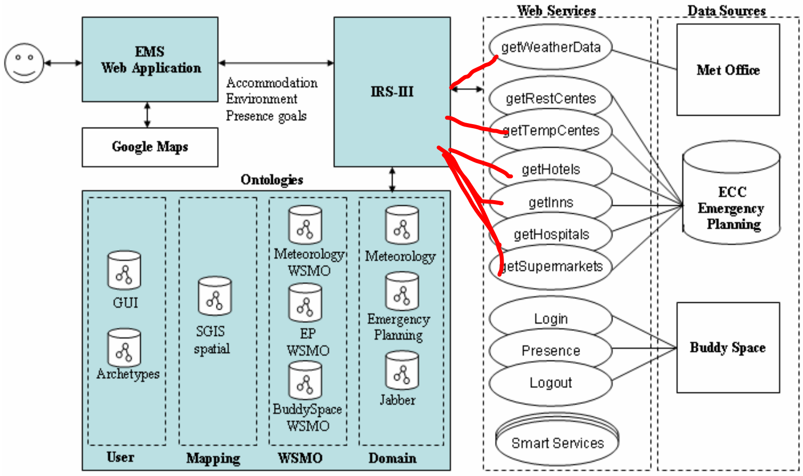

Fig. 2. Architecture of the EMS prototype. Dark boxes represent the main modules of the prototype, white ones are external distributed resources.

Data and functionalities of external information sources are exposed by means of *Web Services* (Section 3.3), semantically described by *Ontologies* (Sections 3.4 and 3.5), and accessible to the EPO through the *EMS Client* (as described in Section 4) which is a web interface using *Google Maps API*. At the heart of the system stands *IRS-III* (Section 2.2). At the moment, the system handles *accommodation*, *environment* and *presence* related goal invocations, discovering SWS that satisfies these goals, managing SWS orchestration and mediation, executing the WS, and returning mediated WS results.

3.2 Data

The EMS aggregates data and functionalities from three different sources:

- *Meteorological Office.* In the UK, it is an official provider of environmental data (e.g. weather forecasts).
- *ECC Emergency Planning.* A collaboration between ECC and British Telecommunications (BT) resulted in the creation and maintenance of a central spatial data repository for the usage of the County, and related agencies such as district councils. In the future it might be made available via the internet to the general public as expected by it[8]. We adopt this repository for SRD retrieval and in particular as source for accommodation information regarding structures that may qualify as shelters during an emergency.

[8] http://technology.guardian.co.uk/weekly/story/0,,1731386,00.html

- *BuddySpace.* An Instant Messaging client built on top of the instant messaging protocol *Jabber*[9] and providing lightweight communication and collaboration means [12]. It allows: (i) presence management, (ii) customizable and interactive graphical visualizations (e.g. maps), (iii) automated contact list generation which facilitates access to a community, and (iv) a high degree of scalability. Additional context information can be pushed or requested from location-aware technology or knowledge of a particular community. Filtering of such contextual information, provided by BuddySpace, allows users or systems to find relevant people (functional role and spatial position) in a given emergency situation, and to easily interact with them through text chat. Interestingly enough in an emergency situation, BuddySpace client interfaces can be accessed using smartphones and other handheld devices.

All data sources and functionalities are described and bublished in IRS as Semantic Web Services.

3.3 Services

We distinguish two classes of services: *data* and *smart*. The former refers to the three data sources described above, and are exposed by means of WS:

- *Meteorological Office services* provide weather information (e.g. snowfall) in specific spatial areas.
- *Emergency Planning services* provide the SRD with information about *primary* and *temporary rest centres*, *hotels*, *inns*, *hospitals*, and *supermarkets*. Each WS requires a query area as input, and return the list of required shelters in that area, together with their properties, such as address, key holder, telephone number, etc. The query area is a circle represented by the centre point (in longitude and latitude) and a radius, but can also by a polygon represented by its edges' coordinates.
- Finally *BuddySpace services* allow the EPO to *connect* to the Jabber network, and retrieve the list of relevant *presences*.

Smart services represent specific emergency planning reasoning and operations on the data provided by the data services. They are implemented in Common Lisp and published by means of IRS-III. In particular, we created *Filter Services* that select SRD responding to emergency-specific requirements (e.g. rest centres with heating system, hotels with at least 40 beds, easy to access hospital, etc.). They capture the EPO selection criteria and protocols. As a result the user retrieves only the most suitable information in a specific situation.

Services communicate with IRS-III through XML/SOAP messages. To get the information up to the semantic level (ontology instances), IRS-III implements a *lifting/lowering* module; by defining specific lifting functions, it is possible to create instances of the relevant ontologies by *lifting* information from the XML output data of WS. Inversely, a *lowering* function allows to create XML data inputs of WS from ontology instances. Through lifting WS results are attached to domain ontologies,

[9] http://www.jabber.org/

then through lowering, they are all converted to an XML format understandable by the interface.

3.4 Ontologies

The following ontologies have been developed to semantically support the EMS SWGIS system. For each ontology we specify where they fit in the SWGIS framework described in Section 2.3:

- *HCI Ontology*: part of the user layer, this ontology is composed of HCI and user-oriented concepts. It allows to further specialize the lowered results on the particular interface which is used (e.g. stating that Google Maps API is used, defining "pretty names" for ontology elements, etc.).
- *Archetypes Ontology*: part of the user layer, this is a minimal ontological commitment ontology which tries to provide a cognitively meaningful insight into the nature of a specialized object; by conveying the cognitive ("naïve") feeling that for example an hospital, as a "container" of people and provider of "shelter" can be assimilated to the more universal concept of "house", which we consider to be as an *archetypal* concept, i.e. based on image schemata and therefore supposed to convey meaning immediately [14]. It is moreover assumed that any client, whilst maybe lacking the specific representation for a specific basic level concept, knows how to display such archetypes.
- *SGIS Spatial Ontology*: part of the mediation layer, it describes high level but common concepts of GIS, such as points, spatial objects with attributes, polygons, and fields.
- *Meteorology, Emergency Planning and Jabber Domain Ontology*: representing the concepts used to describe the services attached to the data sources, such as *snow* and *rain* for Met Office, *hospitals* and *supermarkets* for *ECC Emergency Planning*, *session* and *presences* for Jabber. These are part of the domain ontology layer.

3.5 WSMO Descriptions

WSMO based Goals, Mediators, and WS descriptions of our prototype refer to the Met Office, ECC Emergency Planning, and BuddySpace WS. Goal descriptions are using user ontologies, while Web Service descriptions are linked to domain ones. Finally, mediators link goal and web services of each ontology, solving existing mismatches.

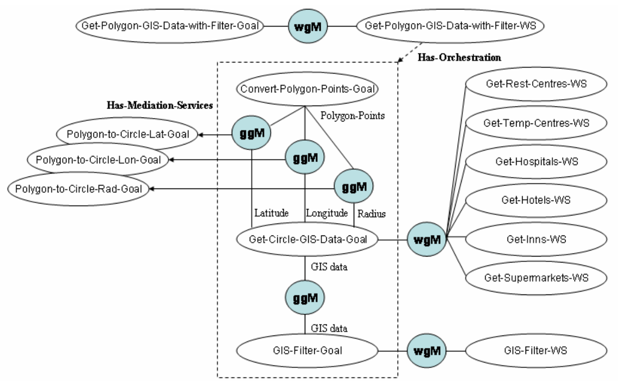

Fig. 3. Structure of the WSMO description of the EMS prototype. To avoid cluttering the diagram, wgM and Web Services balloons were omitted.

To illustrate this interaction we describe in the following (Fig. 3) the structure of the WSMO descriptions associated with one of the goals, *Get-Polygon-GIS-data-with-Filter-Goal*. This goal describes the request of a class of shelter (hospital, inn, hotel, etc.) in a delimited query area. The user (i) specifies the query area through a sequence of at least three points (a *polygon*) before (ii) selecting the requested class of shelter, while ECC Emergency Planning's WS returns the specific class of shelter in a *circular* query area. The results also have to be filtered in order to return only shelter relevant to the specific emergency type (in our case, a snowstorm). The problems are: (1) *selection* of the adequate WS; (2) *mediation* of the different area representations (polygon *vs* circular); (3) *orchestration* of the retrieve and filter data operations. IRS-III offers approaches to solve these problems:

- *WS Selection*: each WSMO description of WS defines, in its *capability*, the specific class of shelter that the service provides. All descriptions are linked to *Get-Circle-GIS-Data-Goal* by means of a unique *wg-mediator (wgM)*. The goal expects as input a class of shelter, and a circular query area. At invocation time IRS-III discovers through the wgM the WS associated to it. Then it selects one amongst them according to the specific class of shelter described in web-service capabilities.
- *Area mediation and orchestration*: *Get-Polygon-GIS-data-with-Filter-Goal* is associated to a unique web service that orchestrates – here, invokes in sequence – three sub-goals. The first one simply gets the list of polygon edges from the input; the second is the above mentioned *Get-Circle-GIS-Data-Goal*; and finally the third invokes the smart service that filters the list of GIS data. The first two sub-goals are linked by means of three *gg-mediators (ggM)* that convert the list of polygon edges provided by the first sub-goal to the centre (latitude and longitude) and radius of the circle that circumscribes that polygon. To accomplish this, we created three mediation services invoked through *Polygon-to-Circle-Lat-Goal*, *Polygon-to-Circle-Lon-Goal*, and *Polygon-to-Circle-Rad-Goal*. The results of the mediation services and the class of shelter are the inputs of the second sub-goal. A unique ggM connects the output of the second to the input of the third sub-goal. No mediation service is necessary.

4 EMS Prototype Interface

The EMS prototype's user interface is web standards based, using *xhtml* and *css* for presentation. *JavaScript* is used to handle user interaction as well as *AJAX* techniques for IRS-III goal invocation. The significant components of the interface are a central *map*, which uses Google Maps API to display *polygons* and *objects* (custom images) at specific coordinates and zoom level. Objects are attached to *goals* and *attributes*, which are displayed in a pop up window or in a hovering transparent region above it.

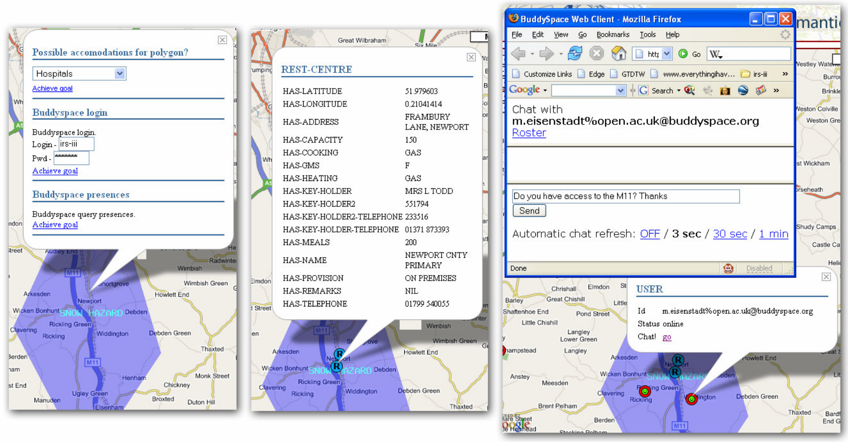

Fig. 4a, 4b and 4c: once defined the area present goals which can be queried to obtain objects and allow further interaction.

As an example of practical usage, we describe how an EPO describes and emergency situation (a *snow hazard* or a *snow storm* each offering different goals), before trying to contact relevant agents. The procedure is as follows:

1. Based on external emergency information the EPO draws a polygon on the map, then assigns a type of emergency to the region. Here, *a snow storm*.
2. Described in an ontology, the new instance has attached *features* and *goals*. Here three goals, one *gets shelters* at distance from the area, two others *connect* to BuddySpace and *get relevant presences*. (Fig. 4a)
3. First, the user requests all *rest centres* inside the region, they are retrieved with their features and attached goals. (Fig. 4b)
4. With that information the EPO *logs* into BuddySpace, then *contacts* the relevant persons to requests action or information. (Fig. 4c)

A screencast of the interaction as well as a live version are available online[10].

[10] http://irs-test.open.ac.uk/sgis-dev/

5 Conclusion and Future Work

In this paper, we describe an ongoing project. Improvements will include adding data dynamic sources (e.g. GPS trackers), extending the ontologies, and verifying that changes integrate naturally at the user level.

However the SWGIS framework we designed and used is operational and proved useful. We believe that this project demonstrates how the Semantic Web - and specifically SWS based systems – can be applied to improve spatial frameworks notably in e-government contexts. Immediate advantages of such an approach are:

1. Automatic selection of the most suitable resources based on current use case.
2. Easing interoperability amongst several SRD providers.
3. Improving the scalability, flexibility, and maintainability of the system.
4. Capturing EPO selection criteria and processes through ontologies for further use.

The final product of this project will be used in ECC. Its usage could be extended to highway agencies, transportation as well as airport authorities. In the long term the SWGIS framework could be opened to citizens in order to provide seamless access to geographic data stored by government agencies.

References

1. Domingue, L. et al. IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. (2004)
2. Gruber, T. R. A Translation Approach to Portable Ontology Specifications. (1993)
3. O. Lassila T. Berners-Lee, J. Hendler. The semantic web. (2001)
4. A. Gugliotta, et al. A Semantic Web Service-based Architecture for the Interoperability of E-Government Services. (2005)
5. R. Casati et al. Ontological tools for geographic representation. (1998)
6. M. J. Egenhofer and D. M. Mark. Naive geography. (1995)
7. D. Peuquet, B. Smith, and B. Brogaard. The ontology of fields. (1999)
8. B. Smith. On drawing lines on a map. (1995)
9. Klischewski, R. Semantic Web for e-Government – a Research Agenda. (2004)
10. Semantic Interoperability Community of Practice (SICoP). Introducing Semantic Technologies and the Vision of the Semantic Web. (2005)
11. F. T. Fonseca and Max J. Egenhofer. Ontology-Driven Geographic Information Systems. (1999)
12. M. Eisenstadt, J. Komzak, M. Dzbor. Instant messaging + maps = powerful collaboration tools for distance learning. (2003)
13. V. Tanasescu, Toward User Oriented Semantic Geographical Information Systems. (2006)
14. D. Mark. Cognitive Image-Schemata for Geographic Information: Relations to User Views and GIS Interfaces. (1989)
15. Agarwal, P. Ontological considerations in GIScience. (2005)
16. Galton. A. A formal theory of objects and fields. (2001)

EGO Ontology Model: law and regulation approach for E-Government

Fernando Ortiz-Rodríguez[1], Boris Villazón-Terrazas[2]

[1]Universidad Valle del Bravo
Calle Laredo 1107, 88760 Reynosa. Tamaulipas. México
fortiz@uvbnet.com
[2]Facultad de Informática. Universidad Politécnica de Madrid
Campus Montegancedo, s/n 28860 Boadilla del Monte. Madrid. Spain
bvillazon@fi.upm.es

Abstract. The Electronic Government is a new field of applications for the semantic web where ontologies are becoming an important research technology. The e-Government faces considerable challenges to achieve interoperability given the semantic differences of interpretation, complexity and width of scope. In this paper we present the results obtained in an ongoing project commissioned by the Spanish government that seeks strategies for the e-Government to reduce the problems encountered when delivering services to citizens. We also introduce an e-Government ontology model; within this model a set of legal ontologies are devoted to representing the Real-estate transaction domain used to illustrate this paper.

1 Introduction and Motivation

Electronic Government (e-Gov) is an important application field [3] for the transformations that governments and public administrations will have to undergo in the next decades. Therefore, to transform the e-Gov into the e-Governance, the e-Gov research needs to be based on a robust theory, on modelling approaches, and on planning. In this scenario, a crucial issue is to manage in different ways the legal knowledge to improve and create semantic systems applications.

The Semantic Web was proposed by Tim Berners-Lee [10] as a new field of research, and according to the World Wide Web Consortium[1] (W3C) the Semantic Web is defined as "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the Web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications".

The application of the Semantic Web to the e-Gov domain is completely new; it features knowledge representation, knowledge engineering, database design, information systems, database integration, natural language understanding,

[1] http://www.w3.org/2001/sw

information retrieval and semantic portals, among others. The Semantic Web is considered to be the infrastructure upon which all intelligent e-Gov applications will be built in the near future. Within the objectives of the Semantic Web the ontologies play an important role.

In the field of the Artificial Intelligence, Neches [15] was the first to define an ontology, and he did it as follows: "Ontology defines the basic terms and the relations that include the vocabulary of a specific area, in addition to the rules to combine terms and relations to define extensions to the vocabulary". Gruber [8, 9] defines the ontology as: "An explicit specification of a conceptualization", being this definition the most referenced in the literature. Borst [1] slightly modify Gruber's definition saying that: "Ontologies are defined as a formal specification of a shared conceptualization". These last two definitions have been merged and explained by Studer and colleagues [20] as follows: "An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group".

In the context of Government and public services, regulation (as a process) is the control of something by rules, accepted by the citizens. The acceptation is made in
last instance indirectly, by their democratic representants in the Parliament by the promulgation of "Laws". Regulation is a compromise between prohibition and no control at all. Public services can encounter conflict between commercial procedures (e.g., maximising profit) and the interests of the people using these services. The Governments have some form of control or regulation to manage this possible conflict. This regulation needs to ensure that a safe and appropriate service is delivered, while not discouraging the effective functioning and development of businesses.

Regulations have several elements, such as [14]:

- Laws or public statutes, promulgated by the Parliaments.
- Reglaments to put in practice the Laws or public statutes by the executive authorities: Governments, different authorities to the Parliaments
- A process of judicial decisions to assure the compliment of the Laws and Reglaments by the citizens, firms or industries and the Governments.

The e-Gov has been strengthened with all these previous studies carried out by the research community and now its main concern is data representation and information management. By its nature, the e-Gov is supported by the legal domain. The legal ontologies for e-Gov applications have been scarce and to reverse this is the first goal of this paper. The second is to build ontologies that help reduce some important semantic problems presented when providing e-Gov services [4].

This research is based on a PhD Thesis and it has been partially prove on a Spanish Project that seeks strategies for e-Gov and aims to provide knowledge conceptualizations (given by legal experts) that help improve information retrieval of legal sources in general and on a Mexican Project that seeks to enhance federal government services at the back office.

This paper is organized as follows: section 2 deals with the related work carried out; section 3 presents the EGO Model; section 4 describes a set of ontologies built on projects, which are part of EGO Model; section 5 describes a couple of systems that use EGO Model. And finally, section 6 is devoted to the conclusions.

2 Related Work

Nowadays the joint efforts put in by different research communities have made possible the birth of the semantic e-Gov. Since e-Gov ontologies are still in their initial state, only a few works carried out in this field are known; thus, in this section we provide a brief state of the art of those works performed in AI, in the law field and in the Semantic Web. The sum up of all these efforts will produce robust ontologies for the e-Gov domain in the near future.

2.1 Law and e-Gov within the Semantic Web

Currently, the Semantic Web is a new area of research and applications within the legal system and e-Gov domains and is a promise for the Web of the next generation; this new area, which is now used mainly to communicate with people but not with machines, will transform the current web since the capability of communication with machines is one of the main objectives of the Semantic Web. If the Web were equipped with more meaning, every citizen would extract answers in a new, easy and simple way and this action could be carried out by web powered semantics, what would enable citizens and businesses to obtain better information from the government. Web powered semantics could help the e-Gov in two ways: first, by allowing the government to delegate more intelligent tasks to computers and second, by solving daily problems with logic deductions and reasoning. But at present, the web is merely a common framework that allows data to be shared and reused.

Currently the legal and e-Gov Semantic Web applications are still in an experimental phase, but their potential impact on social, economical and political issues is extremely significant.

The main goals of e-Gov are to develop user-friendly and efficient services for the public and the business community, though semantic interoperability is also seen as an important issue to solve within this domain. Some of the works aimed at covering the semantic e-Gov domain are the following: the DIP project[2], the Reimdoc project[3], The IFIP Working Group 8.5[4], the Ontogov project[5], the Egov project[6], HOPS project[7], and the WEBOCRAT project[8].

---

[2] http://dip.semanticweb.org
[3] http://reimdoc.atosorigin.es
[4] http://falcon.ifs.uni-linz.ac.at/research/ifip85.html#aim
[5] http://www.ontogov.com/
[6] http://www.egov-project.org
[7] http://www.bcn.es/hops/
[8] http://www.webocrat.org/

2.2 Ontologies: Domain Considerations

The e-Gov scenario is a promising application field for the ontologies underlying the legal engineered knowledge. Many ontologies have been built in the legal domain but not all of them are available or modelled just for a specific domain. The research efforts made in the legal domain by the AI community have contributed to the making of ontologies such as: LLD [12], NORMA [18, 19], FOL [21], FBO [11, 23] and LRI-Core Legal Ontology [2].

The emergence of legal ontologies as part of the Semantic Web initiative has provided a new opportunity for the research community and has brought about a solution to retrieve legal documents within the e-Gov domain. We can mention some of the efforts carried out by AI community on building e-Gov ontologies:

● The Government R&D[9] describes organizations and individuals participating in a government R&D program.

● The Government type[10] describes government concepts used in the CIA World Fact Book 2002.

● The E-Government Ontology[11] describes a seamless UK taxonomy.

3 EGO Model

We use a reference model to focus on and build a common understanding of the problem stated; Figure 1 shows the different actors within the e-Gov.

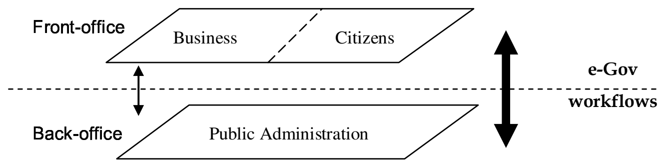

Fig 1. The e-Government Reference Model

At the Back-office, the main actor is the Public Administration; it has many processes inside which should work properly to provide efficient services. The dynamics of the Public Administration provides a huge amount of information to be processed and these data should be managed in a transparent and efficient way.

Within the Public Administration many processes take place and these must be carried out properly to provide efficient services; since the Public Administration functions in a decentralized way and the dynamics of this field generates a huge amount of information to be processed, it is necessary to manage this vast amount of information in a transparent and efficient way. Therefore, the implementation of e-Gov ontologies and applications is crucial.

---

[9] http://www.daml.org/projects/integration/projects-20010811

[10] http://reliant.teknowledge.com/DAML/Government.owl

[11] http://dip.semanticweb.org/documents/D9.3e-Governmentontology.doc

The main objective of designing this initial model in the e-Gov domain is aims to represent the legal issues behind the governments. This model should work as a framework to deploy semantic e-Gov systems, under the law and regulations approach.

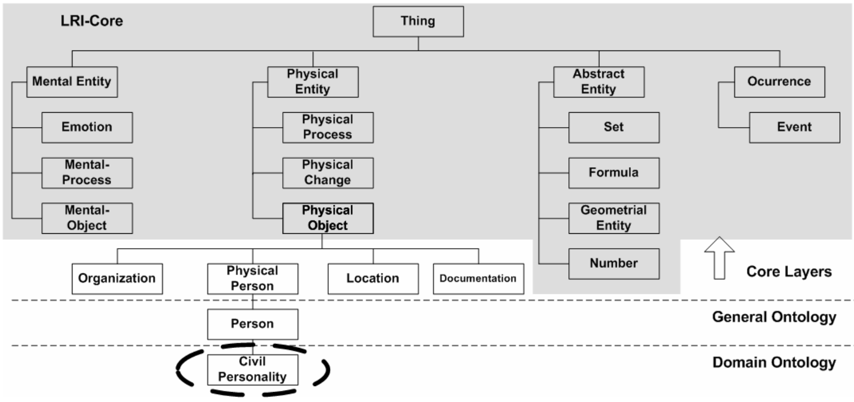

Fig 2. Excerpt of the EGO Ontology Model

The EGO Ontology Model reuses parts of the first two layers of LRI-Core model and is being adapted to the legal system of the Spanish government. The EGO Ontology Model is one of the first efforts not intended for legal domain but for e-Gov domain instead, which is a domain that needs to consider the law, regulations, citizen services, administrative processes, best-practices, and also the different languages spoken within the nation.

3.1 Spanish Case

A particular case is being developed within the Reimdoc[12] Project. This project aims to develop tools that allow the legal document to be modelled in electronic support and be semantically retrieved to facilitate the government-citizen document transaction. The domain selected is related to the Real-estate transaction market and offers sufficient juridical guarantees.

This project will permit verifying the Real-estate processes gathered in digital support. These processes consist of procedures that occur in three areas: the Property Title, the Tributary Administration of the Autonomous Communities and the Justice Administration. In Spain these procedures are meticulously regulated in a coherent form by the context, which is marked by the legal knowledgeable community.

Reimdoc Project is currently developing an application based on the proposed ontologies described in section 5: EgoIR, an Information Retrieval system.

[12] http://reimdoc.atosorigin.es/

3.2 Mexican Case

This project aims to develop the knowledge models necessaries to develop systems that will improve the actual services in federal governments. This ongoing [16, 17] project is reusing actual work done in Spain.

The Tributary Administration (SAT)[13] is including e-Gov ontologies to improve their services given to the front-office (only as a initial effort to include semantic applications in their administration).

The ontologies reused in this project are: person, legislation, organization, civil personality, tax and location. These ontologies are shown in section 4.

The Semantic Web is barely known in Mexico. It is important to begin with little efforts to prove this technology in real environments.

4 E-Government Ontologies

The ontologies described in this section were developed on the Spanish project and are used to illustrate this section.

These ontologies [6] described in this section were built to represent the Real-estate transactions within the Spanish Government domain. These ontologies were developed with knowledge acquired by experts from academic and private sectors and built with the methodology METHONTOLOGY [5] and the workbench WebODE [5].

The ontologies provide support to the EgoIR aforementioned in three important ways: by concept-based indexing, by querying by inference and by improving the navigation. The EgoIR based on these Legal Ontologies bring much focused information, well-defined queries, well-organized information and a sophisticated navigation.

The ontologies presented here are part of an EGO Ontology Model (Figure 2) being develop on this project, this model aims to represent a part of the legal processes carried out within the government.

4.1 EGO Ontology Model Roles

In [13, 22] the five main roles of ontologies are identified: organizing and structuring information; reasoning and problem solving; semantic indexing and searching; semantics integrating and interoperating; and understanding the domain. Before building the Real-estate Transaction Ontologies, we think it should be useful to settle the proper role(s) that the ontology will play.

The EGO Ontology Model (Figure 2) will perform three of the five roles mentioned above: the first role is that of organizing and structuring information in the e-Gov domain, mainly by defining the terms used. The second role is that of reasoning and problem solving; this role basically represents the knowledge of the domain so that an automated reasoner can represent problems and generate solutions

[13] http://www.sat.gob.mx

for these problems, what implies the use of an inference engine to achieve specific goals. The third role is that of semantic indexing and searching (where the ontology will represent the contents of documents) that will enable semantic search for content.

4.2 Reimdoc Project

Figure 3 shows the relationships between the Real-estate Transaction Ontologies aforementioned (each ontology is represented by a triangle). The aim of this figure is to show all the ad-hoc relations between the Real-estate Transaction Ontologies.

For the Reimdoc Project eleven ontologies have been developed: *person*, *civil personality*, *organization*, *location*, *tax*, *contract model*, *jurisprudence*, *Real-estate transaction verifications*, *Real-estate*, *legislation*, and *Real-estate transaction*. Individually, they play the specific goals and model knowledge used in the Reimdoc Project. We describe next the relationships between the main ontologies.

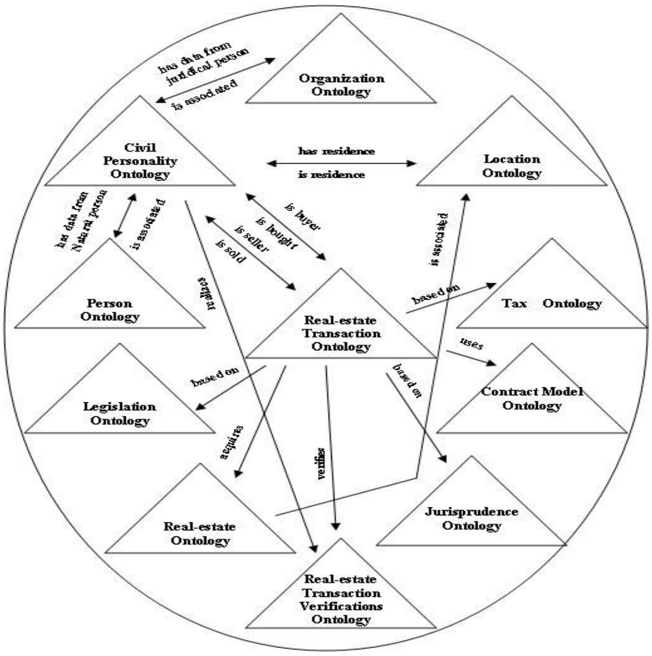

Fig. 3. Main ad-hoc relationships for the Real-estate Transaction Ontologies

The Civil Personality Ontology has as main concept the `civil person`, which is split into two subclasses: `natural person` (representing citizens), `juridical person` (representing enterprises, public administrations, etc.). The ad-hoc relations specified for each concept are those relations whose domain is the concept. For example, the concept `civil person` has six binary relations: 'has data from juridical person', 'has residence', 'is buyer', 'is seller', 'realizes' and 'has data from Natural Person'.

The Real-estate Transaction Ontology has as main concept the `Real-estate transaction`, which is split into two subclasses: `buy` (representing the action of buying), `sell` (representing the action of selling.). The concept `Real-estate transaction` has eight binary relations: 'is bought', 'is sold', 'based on' (tax, legislation, jurisprudence), 'acquires ', 'verifies' and 'uses'.

The Location Ontology has as main concept the `location`, which is split into three subclasses: `geographic division`, `town` and `country`. The concept `location` has two binary relations: 'is residence' and 'is associated'.

The Person Ontology has as main concept the `person`. The concept `person` has one binary relation: 'is associated'.

The Organization Ontology has as main concept the `organization`. The concept `organization` has one binary relation: 'is associated'.

The Real-estate Ontology has as main concept the `Real-estate`. The concept `Real-estate` has one binary relation: 'is associated'.

4.3 Main Ontology Modelling Components

METHONTOLOGY [5] proposes to conceptualize ontologies with a set of tabular and graphical intermediate representations. Such intermediate representations allow modeling the components described in this section.

Concepts are taken in a broad sense. For instance, in the legal domain, concepts are: `Civil Personality, Natural Person, Juridical Person`, etc. Concepts in the ontology are usually organized in taxonomies through which inheritance mechanisms can be applied. For instance, we can represent a taxonomy of legal entities (which distinguishes persons and organizations), where a `Real-estate Contract` is a subclass of a `Contract`, etc.

Relations represent a type of association between concepts of the domain. If the relation links two concepts, for example, `has civil personality`, which links `Natural Person` to `Civil Personality`, it is called binary relation. An important binary relation is *Subclass-Of*, which is used for building the class taxonomy, as shown above. Each binary relation may have an inverse relation that links the concepts in the opposite direction.

Instances are used to represent elements or individuals in an ontology. An example of instance of the concept `Contract` is `Contract of merchanting Real estate`. Relations can be also instantiated. For example, we can express that Contract of merchanting real estate has a location in Madrid as follows: `has location(Contract of merchanting real estate, Madrid)`, using a first order logic notation.

Constants are numeric values that do not change for long time. For example: `legal age`.

Attributes describe properties of instances and of concepts. We can distinguish two types of attributes: instance and class attributes.

Instance attributes describe concept instances, where they take their values. These attributes are defined in a concept and inherited by its sub-concepts and instances. For example, the `date` of a `Contract` is proper to each instance.

Class attributes describe concepts and take their values in the concept where they are defined. Class attributes are neither inherited by the subclasses nor by the instances. An example is the attribute `First Name` as a part of `Natural Person`. Ontology development tools usually provide predefined domain-independent class attributes for all the concepts, such as the concept documentation, synonyms, acronyms, etc. Besides, other user-defined domain dependent class attributes can be usually created.

Formal axioms are logical expressions that are always true and are normally used to specify constraints in the ontology. An example of axiom is a Natural Person has legal capacity at the age of sixteen if he/she gets married.

Rules are generally used to infer knowledge in the ontology, such as attribute values, relation instances, etc. An example of rule is: a Natural Person could be a part of the Juridical Person.

Finally, we present the Real-estate Transaction Ontologies statistics: the number of concepts is 58, the number of relations is 20, the number of attributes is 59, 15 axioms and the number of constants is 2.

5 EGO Model Application Case

We now present two applications that are being developed that employ the proposed EGO Ontology Model. In detail, we present two complementary applications, the P2P system Egoster and the Information Retrieval EgoIR [7]. In general, the two tools differ in their usage perspective and are appropriate for different tasks. However, only the combined application of both tools will offer users the full potential of document management across government.

5.1 EgoIR –Ontology-Based Legal Information Retrieval to Improve the Information Access in e-Government

EgoIR is a java-based system that offers an ontology-based approach to Information Retrieval and its main goal is to retrieve e-Gov documentation. The system deals with government documentation, and gives citizens, business and governments the opportunity to integrate and to recover documents. For this purpose EgoIR provides facilities that manage, search, and share e-Gov documentation. EgoIR also offers an ontology browsing capability (see fig. 4) using the ontologies described in section 4. These ontologies are stored in WebODE [5] (workbench for ontological engineering). Besides, EgoIR allows the construction of a query from the ontology concepts; the query obtained is composed of a set of concepts extracted from the ontologies. EgoIR connects to WebODE throughout WebODE's ODE service to obtain ontology concepts and it employs Lucene12 (search engine library) to retrieve the documents that match the given query. The possibly main users of EgoIR are: a) end users, who require consulting juridical documentation; b) agencies, which need to know the current legislation; and c) lawyers, who have to consult concrete aspects.

Fig. 4. EgoIR User Interface

5.2 Egoster– A Peer-to-Peer System for Sharing Government Documentation

Egoster is a java-based system that exploits semantic web techniques in order to provide an innovative and useful solution for exchanging and retrieving Government Documentation. For this purpose, Egoster provides facilities for managing, searching and sharing Government Documentation in a P2P network, thereby implementing the Model Ontology as a proposal for a standard base to manage Official and Non Official documents across Governments.

Egoster offers a user driven approach where each peer has its own local repository of documents and also has access to the information of others repositories, thus creating a virtual decentralized document repository. The Egoster client on its own (e.g. disconnected from the P2P network) will already provide added value to its users as it will give developers an overview and search facilities of his/her own government documentation stored in its local repository. The goal is: to provide a decentralized Government documentation sharing and retrieving environment using Semantic Web technologies that allows the Back-office (Public Administration) and the Front-office (citizen and business) to interact easily to share documents.

The Egoster is at present time under development as an instance of the Swapster system architecture[14]. It uses ontologies extensively in order to provide some of its main functions importing Government Documentation, formulating queries, routing queries and processing answers.

This system in further development will consider electronic signature and security issues in order to function properly in real environments.

[14] http://swap.semanticweb.org/

6 Conclusions

In this paper we have presented the EGO Ontology Model, even though this model is at initial state, it has well defined goals as supporting semantic applications to retrieve legal documents and, on the other, at delivering services from the public administration (within the government) to citizens. Also we have presented a set of legal ontologies for Real-estate transactions within the Spanish government domain as a part of the EGO Ontology model, which in turn is part of an ongoing project aiming, on the one hand, at supporting semantic applications to retrieve legal documents and, on the other, at delivering services from the public administration (within the government) to citizens. These legal ontologies are built following the methodology METHONTOLOGY and the workbench WebODE and are application independent.

The e-Gov domain does still have many needs: knowledge, for instance, has not been modeled at all. These needs represent real challenges for researchers. One problem to be solved in the near future is that of knowledge acquisition by legal experts. We must add here that the legal domain is very complex and evolving and its complexity provides a different situation than that provided by domains such as physics or mathematics, and this fact will bring about the deployment of future e-Gov ontologies.

We will be focus on further enhancement and evaluation of the EGO Ontology Model; we will be centred on the reasoning capabilities of these Ontology Model; we will continue integrating the law and regulation knowledge captured on the EGO Ontology Model and we will compare the model with other ontology models.

Acknowledgments

We would like to thank our project partners for their helpful comments. Specially thanks to Asunción Gómez-Pérez, Raúl Palma, María del Carmen Suárez-Figueroa, Fernando Galindo, Ana Ibarrola and Ferran Jornet. This work is done partially within the Reimdoc Project which is supported by the Spanish Technology and Science Ministry (Project FIT-340100-2004-22).

This project is partially funded by a scholarship granted through the Program of Teaching Staff Improvement (PROMEP) at the Tamaulipas University.

This product includes software developed by the Apache Software Foundation (http://apache.org/)

References

1. Borst, W.N.: Construction of Engineering Ontologies. University of Tweenty. Enschede, NL-Centre for Telematica and Information Technology. (1997)
2. Breuker, J.: Constructing a Legal Core Ontology: LRI-Core.University of Amsterdam, Leibniz Center for Law, Amsterdam, the Netherlands (2004)
3. Curtain, G., Sommer, M., Vis-Sommer, V.(editors): The World of E-Government. The Haworth Press. (2003)

4. Electronic Government: First International Conference, EGOV 2002, Aix-en-Provence, France, September 2-5,. Proceedings (Lecture Notes in Computer Science) by Roland Traunmüller (Editor), Klaus Lenk (Editor) (2002)
5. Gómez-Pérez, A.; Fernández-López, M.; Corcho, O. Ontological Engineering. Springer Verlag. 2003
6. 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 (2005) To appear in CAEPIA'05 post-proceedings volume. Springer Verlag 2006
7. Gómez-Pérez, A., Ortiz-Rodríguez, F., Villazón-Terrazas, B:.Ontology-Based Legal Information Retrieval to Improve the Information Access in e-Government. To appear in WWW2006, Edinburgh, Scotland (2006)
8. Gruber, T.R.: A Translation Approach to Portable Ontologies. Knowledge Acquisition. 5(2) (1993) 199-220
9. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Presented at the Padua workshop on Formal Ontology, to appear in an edited collection by Nicola Guarino (1993)
10. Hendler, J., Berners-Lee, T., and Miller, E. Integrating Applications on the Se-mantic Web, (2002) http://www.w3.org/2002/07/swint.html
11. Kralingen, R.W.: Frame-based Conceptual Models of Statute Law, Computer/Law Se-ries. No.16, Kluwer Law International. The Hague, the Netherlands (1995)
12. McCarty, L.T.: A Language for Legal Discourse, I. Basic Features, Proceedings of the Second International Conference on Artificial Intelligence and Law, Vancouver, Canada (1989) 180-189
13. Melz, E., Valente, A.: Modelling the Tax Code. The Second International Workshop on Regulatory Ontologies (2004)
14. Montesquieu, C.: El espíritu de las leyes. Imprenta de D. Marcos Bueno. Madrid (1845) vol I, book XI, cap. VII 189-201
15. Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., Swartout, W.R.: Enabling Technology for Knowledge Sharing. AI Magazine. 12(3) (1991) 36-56.
16. Ortiz-Rodríguez, F.: Mexican E-Government Ontologies: an adaptation. In Procedings of the Fourth Latin American and Caribbean Conference. Mayaguez, Puerto Rico (2006)
17. Ortiz-Rodríguez, F.: Ontologías de E-Gobierno para Gobiernos Federales. Memorias en extenso de el 18 Encuentro Nacional de Investigación del Golfo de México (2006)
18. Stamper, R.K.: LEGOL: Modelling Legal Rules by Computer. Computer Science and Law. Bryan Niblett. Cambridge University Press. Cambridge, United Kingdom (1980)
19. Stamper, R.K.: The Role of Semantics in Legal Expert Systems and Legal Reasoning. Ratio Juris, Vol. 4, No. 2 (1991) 219-244.
20. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering. 25: 161-197. (1998)
21. Valente, A.: Legal Knowledge Engineering; A Modelling Approach. University of Amsterdam, The Netherlands, IOS Press, Amsterdam, The Netherlands (1995)
22. Valente, A.: Types and Roles of Legal Ontologies. V.R. Benjamins et al. (Eds.): Law and the Semantic Web, (2005) 65-76
23. Visser, P.R.S.: Knowledge Specification for Multiple Legal Tasks; A Case Study of the Interaction Problem in the Legal Domain, Computer / Law Series. No. 17, Kluwer Law International. The Hague, The Netherlands (1995)

Mapping Citizen Profiles to Public Administration Services Using Ontology Implementations of the Governance Enterprise Architecture (GEA) models

Sotirios K. Goudos[1], Vassilios Peristeras[1], Konstantinos Tarabanis[2]

[1]Informatics and Telematics Institute, Center for Research and Technology Hellas,
1st km Thermi-Panorama Road, Thermi, 57001,
Thessaloniki, Greece,
{sgoudos, per}@uom.gr

[2]University of Macedonia
Information Systems Laboratory
Thessaloniki, Greece
kat@uom.gr

Abstract. The problem of matching a citizen's needs with available public services is discussed in this paper. This work is based on the Governance Enterprise Architecture (GEA) object models and semantic web technologies. An ontology is created based on the above object model. This ontology is then used as the knowledge base for a semantic application. The input to the application consists of the user profile, while the output returned consists of the public services that match the specified profile.

1 Introduction – Motivation

A dual communication/integration problem in contemporary public administration has been identified and discussed in [1], that is:

- internally among public administration (PA) agencies, resulting in the creation of stovepipe systems with minimum horizontal information flows;
- externally between PA and its external environment, resulting in out-of-date, inadequate and frustrating citizen-PA communication.

Due to these inconsistencies, a clear business need emerges for all PA systems to develop advanced internal and external interfaces to address this dual PA integration deficit; that is, (a) to achieve internal integration at the administrative intra- and inter-agency level, as well as (b) external integration and user-centric communication channels with society.

The work presented here contributes to the second type of the above-presented PA integration deficit. It does so, by implementing an application that facilitates the identification of relevant PA services based on the profile of a citizen[1]. This profile-to-service resolution is related to the more general problem, that of mapping the citizen needs to PA services as provided by each public administration system [2]. Briefly, the problem stems from the different perspectives PA and citizens hold for PA services.

Initially the citizen has a need. He/she may not know which public services are currently available by public administration to address this need. The citizen is needs-aware, but not services-aware (being aware of the actual services he/she really needs). On the contrary, PA is services-aware, but not needs-aware. The need may arise for several reasons, e.g. due to an external event (life-event or business episode) or due to the profile of the customer (e.g. unemployed with five children). This later case attracts our interest in this paper.

The services that address this need may be mandatory (e.g. to register a new-born child) or simply beneficial to the citizen (e.g. receive a grant). To facilitate the communication between the two actors, there is a need for a consistent mapping between services and needs and vice-versa.

Usually this matching is done on an ad-hoc basis by the citizen alone in an empirical, time and energy consuming, as well as frustrating way, without much support from public administration. Taking into account the fragmentation of the administrative space this task is currently performed in a very suboptimal manner.

To facilitate this task, we introduce a conceptual component, which we call *Needs-to-Services Converter*. This receives a citizen's need as input and provides as output a set of public administration services that address this need. The existence of such a converter is clearly a business need for the smooth operation of PA and could be implemented using different technologies.

In this paper, we try to implement part of this *Needs-to-Services Converter* - more specifically the profile-to-services mapping - by employing semantic technologies and using a PA service model, as introduced by the Governance Enterprise Architecture (GEA).

This paper is organized as follows: An overview of GEA and the GEA service model is presented in Section 2. The prototype system is presented in section 3. In Section 3.1, the business case is described. The ontology implementation of the GEA model in OWL is given in Section 3.2. Sections 3.3 and 3.4 present respectively the proposed system architecture and the running prototype system. Finally conclusions and plans for future work can be found in Section 4.

2 The Governance Enterprise Architecture

GEA aims at introducing a consistent set of models that constitute the basis for reference eGovernment domain ontology. This ontology is generic enough to cover the overall eGovernment domain, and at the same time specific enough to sufficiently model PA specific semantics. A key aspect of GEA is that it attempts to be technol-

[1] By citizen, here we mean any type of PA clients (e.g. business, other legal entity, citizen)

ogy-neutral. This means that the GEA models may be applicable to different technological environments. A GEA overview can be found in [2]. The models are presented in detail, in [3-9].

For the purpose of this paper, we focus on the GEA detailed object model for service provision referred to in this paper as the PA Service Model for the sake of brevity. This stems from the fact that it is this model that is directly linked to the representation of a PA service, and thus is most relevant to the work of this paper.

The overall model [9] is presented in Fig. 1. A brief textual description follows.

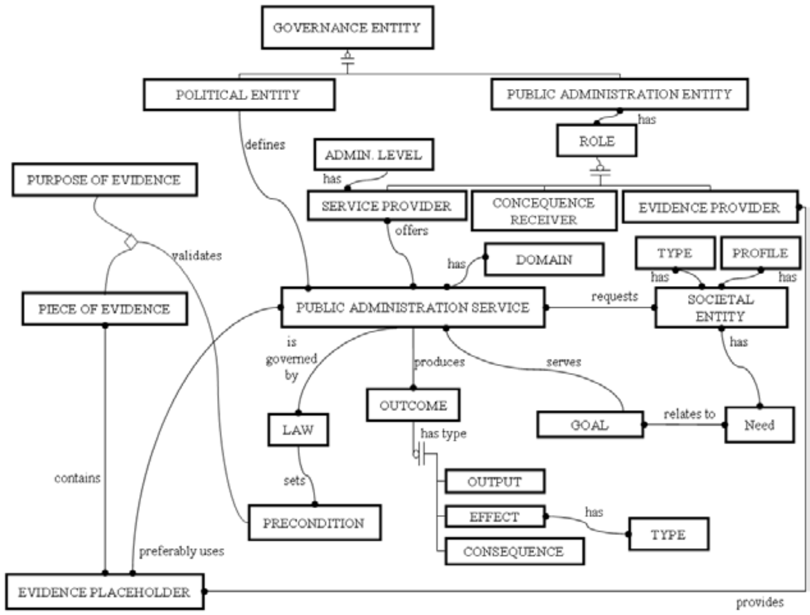

Fig. 1. The GEA PA Service Model

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) and each Social Entity has a *Profile* (e.g. young businessman, disabled person).

There are two categories of *Governance Entities* participating in service provision: *Political Entities* and *Public Administration Entities*. Based on the role which PA Entities can acquire during the service execution phase, we identify three roles:

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

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. The m:n relationship between the two entities expresses the fact that specific Evidence can be found in numerous different Evidence Placeholders. For example, a citizen's age, serving as a Piece of Evidence for a service that sets age limitations in its Pre-conditions, can be contained in the ID card, the passport or the birth certificate. These are considered as alternative Evidence Placeholders. There are many cases where the Evidence Placeholders are provided by PA Entities (Evidence Providers).

The direct relationship between PA Service and Evidence Placeholder depicts cases where PA Services preferably use specific types of Evidence Placeholders, e.g. when the law explicitly states that a birth certificate is needed for the execution of a particular service.

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

Output – is the documented decision of the Service Provider regarding the service asked by a Societal Entity. This "documented decision" is currently embedded and reaches the client in the form of an administrative document/decision.

Effect – the execution of a service may result in a change in the state of the world (e.g. transfer money to an account). In the PA domain, the service Effect is the actual permission, certificate, restriction or punishment the citizen is finally entitled to. In cases where administration refuses the provision of a service, there is no Effect. At the top level, there are three *types* of Effects expected from the execution of PA services. These have been identified to be the following:

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

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

In conclusion, from a Model Driven Development (MDD) approach, we may say that the GEA PA Service Model is a "… Computational Independent Model (CIM) describing the business context and business requirements" [10]; in our case, the PA context and requirements.

3 Prototype Description

The business case of the needs-to-services task was described in the introduction. In our case the system should be able to find all the public services that are suitable for a certain user profile. An additional element is that of the citizen's residence area. The citizen should be able to identify all the public services suitable for him/her based only on his/her profile description and are available in his/her residence area. This information is available not only for humans, but also for software applications. In a second stage the application should be able not only to discover, but also to execute some of the public services found in this way. This problem is similar to the well-known wine agent demo [11], which is used as the main example in OWL Recommendations [12].

3.1 Use Case

When the user enters the first page of the application, a list box containing a list of client profiles, appears. The user selects a profile that he/she believes is best suited for him/her. At this point various queries are sent to a reasoner asking for the service properties relevant to this profile. When all the properties are known, the reasoner creates a new query that searches for public services with the properties found in the previous queries. Finally, the reasoner returns the services that were found in a new web page. The system uses JSP (Java Server Page) and Java servlets in order to execute the user request and dynamically create the results in HTML format respectively.

3.2 The GEA model Ontology

The GEA object model was shown in an E-R diagram. For simplicity, the model was not shown in its entirety. The model implementation was the main issue that concerned us. It is obvious that such a model can be implemented using a relational database. Such an approach would be complex since it would not exploit the advantages of declarative knowledge representation. The main requirement today is to be able to share information through the web for both humans and machines. We decided to express the GEA model in an ontology language.

OWL DL [12] was the obvious choice for several reasons; since 2004, OWL DL is an active recommendation of W3C group and various examples of models expressed in OWL exist. Several one-to-many relations exist in the GEA Service model. Therefore, the full expressiveness of cardinality restrictions requires the use of OWL DL instead of OWL Lite. Another important point that was taken into account is the existence of OWL DL reasoners. OWL Full [12] reasoners do not yet exist. The GEA ontology has been created using the Protégé tool with the OWL plug-in [14,15].

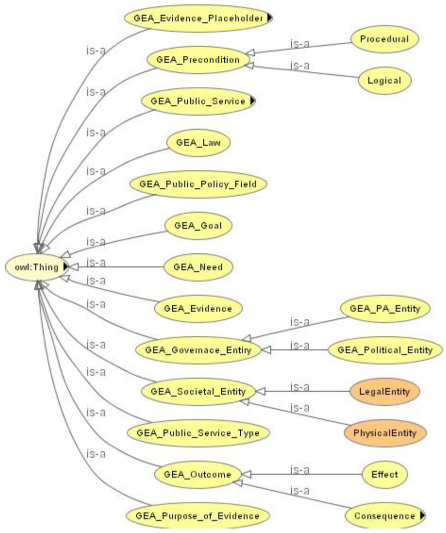

Fig. 2. The GEA service ontology class hierarchy

A part of the GEA model class hierarchy in OWL DL is shown in Fig. 2. The basic modeling principles followed were:

1) The GEA model entities were expressed in `owl: Class` elements
2) The relations between entities were expressed in `owl : ObjectProperty`. Metaclasses. In cases were the relations were one-to-one, they were expressed in `owl: FuctionalProperty` metaclasses.
3) It is known that OWL DL uses the Open World Assumption (OWA) so all classes at the same hierarchical level and all individuals that belong to same class have to be explicitly declared as different. Therefore all the classes in the same hierarchy level were declared `owl: disjointWith`. The individuals that belonged to the same class were declared `owl: Alldifferent.`
4) In the GEA object model several has-type relations exist between entities. In some cases these were modeled in OWL as `rdfs: subClassOf`. In some other cases these relations were modeled using object properties. For example, the Output object was not modeled as a subclass of the Outcome object. This design decision that lead to this was the fact that the Output object of the GEA model represents a documented decision therefore an Evidence-Placeholder. Two new object properties were created; `hasOutput` with domain `GEA_Public_Service` and range `GEA_Evidence_Placeholder`, and its inverse property `isOutputOf`. The `Effect` and `Consequence` objects were modeled as subclasses of the `Outcome` class.

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

The classes that were not shown in Fig. 2 are given in Fig. 3. These classes represent the property descriptors of the `GEA_Public_Service` class. Due to visualization reasons only the classes that are used in the application appear. Below follows a brief description of these classes.

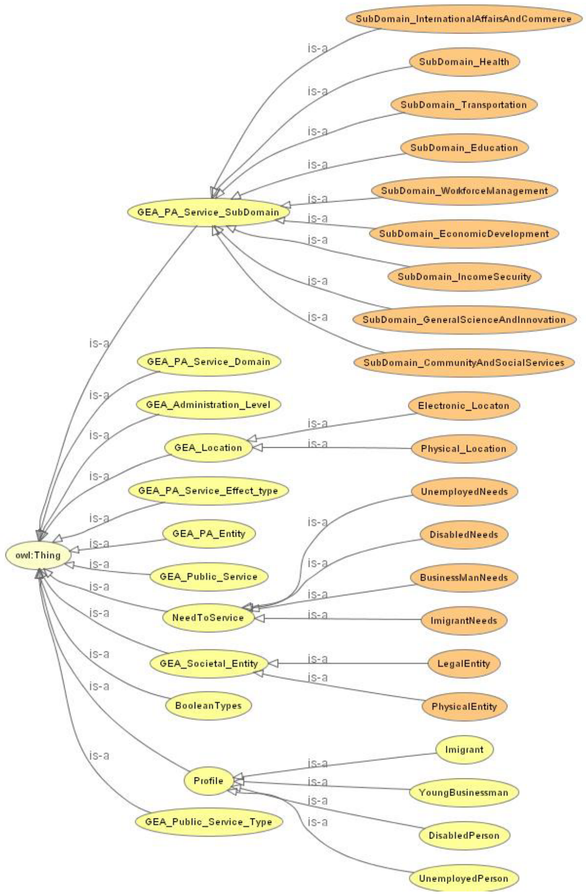

Fig. 3. The GEA ontology class hierarchy with GEA_Public_Service descriptors classes.

`GEA_PA_Service_Domain` class, which represents the different PA domains. It has been populated with the following individuals; `CommunityAndSocialServices`, `EconomicDevelopment`, `Education`, `Health`, `Transportation`, `GeneralScienceAndInnovation`, `IncomeSecurity`, `Interna-`

tionalAffairsAndCommerce and WorkforceManagement. These individuals are declared different using the owl:all Different element. One may notice that these domains correspond to USA Federal Enterprise Architecture (FEA) [16] Business Reference Model ServicesforCitizens object.

Each of these main domains has a number of subdomains. These are represented by the GEA_PA_Service_SubDomain Class (equivalent to SubFunction object in FEA [16]). Subclasses of this class have been created and all are modeled as an owl: EquivalentClass. Each subclass represents a PA domain and the individuals who belong to the class represent the PA SubDomains. For example, the definition of the SubDomain_Transportation Class is given in Fig. 4 using OWL abstract syntax [13]. The GEA_Administration_Level class is populated by four individuals Ministry_Level, Prefecture_Level, Municipality_Level and Region_Level. This class represents the unique administration level at which each public service is offered.

```
Class(SubDomain_Transportation complete restriction(isSubDomainof value(Transportation)))

SubClassOf(SubDomain_Transportation GEA_PA_Service_SubDomain)
```

Fig. 4. The SubDomain_Transportation class in OWL abstract syntax [13].

The GEA_Location represents the physical or electronic location where the public service is offered. For the physical location, a top-level, location ontology can be imported. The GEA_Public_Service_Effect_Type class represents at an abstract and high-level, the three distinct effect types achieved by a public service. The individuals that belong to that class are ObtainSustainableDevelopment, PromoteSocialWelfare and SafeguardSocialContract.

All the above public service descriptors are linked with individuals from the GEA_Public_Service class using owl:ObjectProperty elements with domain GEA_Public_Service and range the corresponding public service descriptor class. The GEA_Public_Service Class declaration is shown in Fig. 5

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

Fig. 5. The GEA_Public_Service in OWL abstract syntax.

The individuals that belong to GEA_Public_Service class are public services. For example the public service individual that corresponds to the issuance of a parking license for a disabled person is shown in OWL abstract syntax in Fig 6.

```
Individual(DisabledParkingLicenseIssuance annotation(rdfs:comment "A service that issues  a parking license that is free for disabled
persons"^^<http://www.w3.org/2001/XMLSchema#string>)
                                          type(GEA_Public_Service)
                                          value(hasPADomain Transportation)
                                          value(hasClientType Citizen)
                                          value(hasPublicServiceType GEA_Certification)
                                          value(hasAdministrationLevel Municipality_Level)
                                          value(hasEffectType PromoteSocialWelfare)
                                          value(hasPASubDomain GroundTransportation))
```

Fig. 6. The DisableParkingLicenceIssuance individual in OWL abstract syntax.

```
Class(DisabledNeed complete restriction(hasPerson allValuesFrom(DisabledPerson))
                            NeedToService)

SubClassOf(DisabledNeed restriction(hasService allValuesFrom(restriction(hasClientType value(Citizen)))))
SubClassOf(DisabledNeed restriction(hasService allValuesFrom(restriction(hasEffectType value(PromoteSocialWelfare)))))
SubClassOf(DisabledNeed restriction(hasService allValuesFrom(restriction(hasAdministrationLevel value(Municipality_Level)))))
SubClassOf(DisabledNeed restriction(hasService someValuesFrom(restriction(hasPADomain value(CommunityAndSocialServices)))))
SubClassOf(DisabledNeed restriction(hasService someValuesFrom(restriction(hasPADomain value(Transportation)))))
```

Fig. 7. The DisabledNeed class in OWL abstract syntax.

The three main classes that play an important role in the application are: the `Profile` class, the `GEA_PublicService` class and a new class which we call `NeedToService` class. The `Profile` class represents the different user profiles (e.g. immigrant, unemployed). For the needs of the prototype only four subclasses of the above class were created. These represent the client profiles of a young businessman, an immigrant, a disabled and an unemployed person. The NeedToService class (equivalent to MealCourse class in the wine ontology) links the profiles with services. In Fig. 7 the DisabledNeed class is given in OWL abstract syntax.

3.3 System Architecture

The proposed system architecture is given in Fig. 8. It consists of a web server, a reasoner and an OWL file which is used as the knowledge base. The users access the application through a common Internet browser. The advantage of every web-based front-end is that it requires only an Internet browser in order to execute and it can be accessed from anywhere on the Internet. The system architecture employed is server-side; therefore the client shows only the form and the results page.

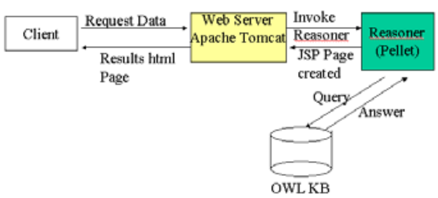

Fig. 8. The application architecture

The server uses the data given to invoke the reasoner. The reasoner sends various queries to the knowledge base. SPARQL [17] was selected as the query language. The answers returned are parsed by the web server that creates the results web page.

The extracted results contain a list of the public services that match the selected profile. Specifically, the web server used was Apache Tomcat, the knowledge base was an OWL file with the GEA ontology. The reasoner selected is Pellet [18]. Pellet is an open source OWL DL reasoner that can be used in conjunction with Jena either through a DIG interface or with native APIs. Pellet provides support for SPARQL.

3.4 Running example

We consider the public service of issuance of a license for parking places near someone's residence area as an example for our application. The residents that need a parking license may apply to the Municipality of their area given the fact that they have to pay a monthly fee. Residents that are disabled do not have to pay any fee. Therefore people with different profiles will have access to different public services. The definition of the public service for residents is similar to that of the disabled parking license service shown in Fig. 8, with one difference, the `gea:hasEffectType` property has the value `gea:ObtainSustainableDevelopment`. Let us assume that the user selects the disabled person Profile. A new individual of the class `DisabledNeed` (subclass of `NeedToService` class) is created on the fly. Then SPARQL queries are sent to the reasoner (Pellet) asking for the properties of the public services that are linked to this individual. Such a SPARQL query asking for the value of `hasPADomain` property is given in Table 1 below.

Table 1. SPARQL query for finding the value of hasPADomain property.

```
PREFIX gea:  < http://localhost/GEA.owl #>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT  ?Domain
WHERE { ? NeedToService rdf:type gea: NeedToService.
        ? NeedToService gea:hasService ?Service.
        ?Service gea:hasPADomain ?Domain.}
```

Table 2. First SPARQL query for public services that match the found properties.

```
PREFIX gea:  < http://localhost/GEA.owl #>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT  ?Service
 WHERE {
 ?Service rdf:type gea:GEA_Public_Service.
 ?Service gea:hasPADomain gea:Transportation.
 ?Service gea:hasEffectType gea: PromoteSocialWelfare.
?Service gea:hasAdministrationLevel gea: Municipality_Level.
?Service gea:hasClientType gea: Citizen.}
```

The reasoner returns two PADomain values: `gea: CommunityAndSocialSer-vices` and `gea:Transportation`. Similar queries are performed for all `GEA_Public_Service` properties. When all the public services properties are known, then we ask the reasoner to find all the public service individuals that match the found properties. Due to the fact that two PADomain values are found, two queries are performed. The first is given in Table 2. A similar query is executed for `gea: CommunityAndSocialServices` PADomain. After query execution the reasoner then returns the public services `gea: DisabledParkingLicenseIs-suance` and `gea: DisabledBenefitIssuance`.

4 Conclusion and future work

The GEA object model for public services can be used as the basis for a reference eGovernment domain ontology. A GEA OWL DL ontology based on this model was created. This ontology serves as the knowledge base for the implementation of a Needs-to-Services semantic application. The system concept is similar to that of the wine agent demo. The user selects his/hers profile and the application uses reasoning to find the matched public services. The GEA object model is proven to have the adequate expressiveness for such an application.

This application is the first step towards semantic public service discovery and execution. An approach based on a Semantic Web Services framework will be the next step. Another issue that is not addressed in this paper is the storage of the ontology in a repository. A large number of public services exist in public administration therefore their ontological representation and storage requires an efficient and proven technology. Performance issues are also of great importance for every reasoner and repository. Semantic interoperability in a pan-European level is also a problem that can be solved using the GEA ontology model. These issues will be part of our future work.

Acknowledgements

This work has been partially funded by the “SemanticGov” EU project (FP6-2004-IST-4-027517).

5 References

1. Peristeras V., *The Governance Enterprise Architecture - GEA - for Reengineering Public Administration, PhD Dissertation*, in *Business School Department*. 2006, University of Macedonia: Thessaloniki.
2. Peristeras V. and Tarabanis K. *Reengineering the public administration modus operandi through the use of reference domain models and Semantic Web Service tech-*

nologies. in *2006 AAAI Spring Symposium, The Semantic Web meets eGovernment (SWEG)*. 2006. Stanford University, California, USA: AAAI.

3. Peristeras V. and Tarabanis K., *The Governance Enterprise Architecture (GEA) Object Model*, in *Knowledge Management in Electronic Government, Lecture Notes in Computer Science 3035, (5th IFIP International Working Conference, KMGov 2004, Krems, Austria, May 27-29)*, M.A. Wimmer, Editor. 2004, Springer. p. 101-110.

4. Peristeras, V. and K. Tarabanis, *Towards an Enterprise Architecture for Public Administration : A Top Down Approach.* European Journal of Information Systems, 2000. **9**(Dec. 2000): p. 252-260.

5. Tarabanis, K., V. Peristeras, and G. Fragidis. *Building an Enterprise Architecture for Public Administration: A High Level Data Model for Strategic Planning*. in *9th European Conference on Information Systems*. 2001. Bled, Slovenia.

6. Tarabanis K. and Peristeras V., *Knowledge Management Requirements for Pan-European Public Administration Service Delivery*, in *Knowledge Management in Electronic Government: 4th IFIP International Working Conference, KMGov 2003, Rhodes, Greece, May 26-28, 2003. Proceedings*. 2003, Springer-Verlag Heidelberg: Rhodes Island, Greece.

7. Peristeras, V., T. Tsekos, and K. Tarabanis. *Building Domain Models for the (e-) Governance System*. in *International Conference on Politics and Information Systems: Technologies and Applications (PISTA '03)*. 2003. Orlando, Florida, USA.

8. Peristeras V. and Tarabanis K. *The GEA Generic Process Model for Public Administration Service Execution*. in *8th International Conference for Business Information Systems (8th BIS 2005)*. 2005. Poznan, Poland.

9. Peristeras V. and Tarabanis K., *Advancing the Government Enterprise Architecture - GEA: The Service Execution Object Model*, in *Electronic Government, DEXA, 3rd International Conference EGOV 2004, Zaragoza, Lecture Notes in Computer Science 3183*, R. Traunmuller, Editor. 2004, Springer. p. 476-482.

10. ATHENA Project, *D.A6.1, Specification of a Basic Architecture Reference Model*. 2005.

11. Wine agent demo, Stanford University, <http://onto.stanford.edu:8080/wino/index.jsp>

12. Dean M., Schreiber G., Bechhofer S., van Harmelen F, Hendler J., Horrocks I., McGuinness D., Patel-Schneider F., and Stein L.A.: OWL web ontology language reference, W3C Recommendation, (2004).

13. Patel-Schneider, P.F., Hayes, P., Horrocks, I. (eds.): OWL Semantics and Abstract Syntax. W3C Recommendation, (2004).

14. The Protégé Ontology Editor, Stanford Medical Informatics, Stanford University School of Medicine, <http://protege.stanford.edu/>.

15. Knublauch Holger, Fergerson Ray W., Noy Natalya F., Musen Mark A.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications, Third International Semantic Web Conference - ISWC 2004, Hiroshima, Japan, (2004).

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

17. A. Seaborne, E. Prud'hommeaux, SPARQL Query Language for RDF, W3C Working Draft (work in progress), 19 April 2005, Latest version available at http://www.w3.org/TR/rdf-sparql-query/

18. Pellet OWL Reasoner available at http://www.mindswap.org/2003/pellet/

Interoperability in eGovernment through Cross-Ontology Semantic Web Service Composition

Nils Barnickel, Matthias Fluegge, and Kay-Uwe Schmidt

Fraunhofer Institute for Open Communication Systems,
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{Barnickel, Fluegge, Schmidt}@fokus.fraunhofer.de

Abstract. Due to the heterogeneous structure of the public sector the achievement of interoperability is a key challenge for comprehensive electronic Government. Service oriented architectures lay the foundation for flexible application integration and process-orientation through Web service composition. Semantically enriched Web services promise to increase the level of automation and to reduce integration efforts significantly. In this paper we present an approach for semi-automatically supporting the design of data flows between semantically described Web services which are making use of different ontologies and data representations. The approach includes a rule-based mechanism for user-transparent mediation between ontologies. In order to validate the approach we have implemented a prototypical Cross-Ontology Semantic Web Service Composition Tool to be used in eGovernment scenarios spanning multiple application domains.

1 Introduction

Today XML-based standards such as WSDL, SOAP and BPEL are widely used for describing, composing and invoking Web services. These technologies represent the foundation for establishing syntactical interoperability between different applications. The necessary steps for composing services are primarily done manually. In a first step appropriate services have to be selected from hierarchical repositories[1] whereby the specific category for each service needs to be known a priori. Having chosen appropriate services a user needs to understand their implicit semantics in order to design the control flow and the data flow.

With respect to the multitude of services participating in a process the data flow modeling, i.e. the parameter assignments between the activities, may be a time-consuming task and it requires the user to have and extensive knowledge about the underlying type representations.

In particular, when composing services from different application domains (e.g. citizen registration and vital records) comprehensive data type transformations have to be added manually due to the existing different data representations in different domain standards.

[1] UDDI, ebXML

The idea of bringing implicitly defined service semantics to an explicit level by providing machine understandable Web service descriptions with formally defined semantics promises to support the composition process. The long term vision is to enable dynamic goal-oriented service composition and to use powerful inference engines and matchmaking mechanisms in order to automate the whole composition process including discovery, composition, execution and interoperation of Web services. As it has been argued in [FST2006] on the road towards this goal still many problems need to be solved whereby each further step can increase the level of automation.

Electronic Government is an ideal testbed for Semantic Web research due to the heterogeneity of information space with the challenge to achieve interoperability and process integration. At the same time the eGovernment domain exhibits a high degree of formality in key areas imposed by laws thus encouraging the application of Semantic Web technologies based on formal modeling and description logics. Semantic Web technologies represent the foundation for achieving the vision of a knowledge-based, user centric, distributed and networked eGovernment [Mg2006].

2 Integration of eGovernment Services

EGovernment is the use of information and communication technology to promote more efficient and user-friendly public services. Due to the heterogeneous and distributed nature of the eGovernment domain the integration of application is a crucial issue. Providing public services often involves a multitude of public agencies which are using highly-specialized applications. Service oriented architectures in general and Web service technologies in particular are the foundation for flexible application integration and for implementing processes spanning multiple organizational domains and applications.

Today, in order to ensure interoperability, eGovernment applications provide standard Web service interfaces including well-defined message sets. In fact, in various countries national interoperability frameworks impose XML schemes and Web service interfaces for exchanging data between administrations. An example for such a national effort is the Danish eGovernment initiative which focuses not just on the definition but also on the reuse of base types and XML domain data structures [BN2003]. A key achievement of the initiative is the "InfoStructureBase"[2], a shared repository for the XML-based schemas. In Germany due to its federal structure the approach is less centralized. Although there are some initiatives like OSCI-XÖV[3] there is just a limited central control. Furthermore interoperability is just ensured within domain boundaries, e.g. through OSCI/XMeld (information exchange between registration offices) or OSCI/XJustiz (XML Schema exchange standard for legal authorities), but these standards do not assure cross-domain interoperability. Even more difficult than establishing XML standards for one country is to achieve interoperability

[2] http://isb.oio.dk/info

[3] http://www.osci.de

between different countries. In cross-organizational and cross-border eGovernment processes services of various public agencies from different domains and with different areas of operations are involved. In such scenarios the lack of semantic interoperability results in enormous integration efforts.

The online order of a birth certificate, as illustrated in figure 1, demonstrates the integration problem in a cross-organizational scenario. The process includes a service for handling the payment of the birth certificate fee, a resident registry service for checking the citizen input for consistence, a vital records office responsible for issuing the birth certificate, and a statistical office to which the vital records office reports its activities. Assuming the lack of a digital signature infrastructure the output of the birth certificate order is still a paper-based birth certificate.

In the given scenario the domain standard employed by the resident registry uses a different data representation for names and addresses than that used by the vital records office. While in the one domain an address might be a complex type consisting of different attributes for given name, surname, street, street number, etc. and in the other domain standard the address concept might be modeled as a complex type that contains just one single attribute for street and street number all together.

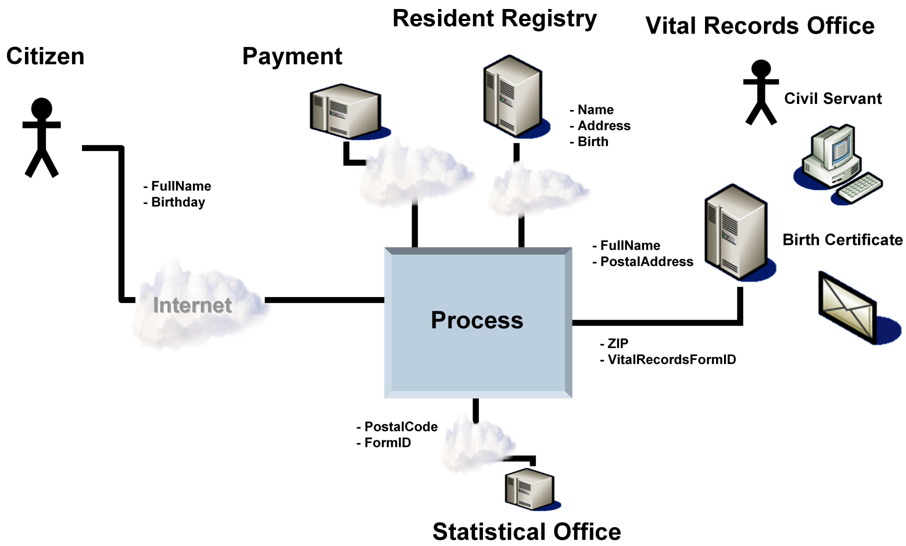

Fig. 1. Internet order of a birth certificate.

It is a fact that most eGovernment data exchange standards are being developed independently from each other and that in different eGovernment application domains the requirements for information granularity differ significantly. Therefore it is not feasible to address this problem by introducing a global ontology or a global schema. In order to serve best for intra-domain integration domain standards need to evolve independently from each other.

In the following chapter we present a composition approach that provides a mechanism to ease semantic interoperability for inter-domain integration while at the same time preserving the independence of domain-specific standards. The presented concepts are based on the research efforts undertaken in the SATINE[4] project. As part of this project an architecture and a toolkit for supporting semi-dynamic service composition in a semantic-based interoperability infrastructure have been developed [FT2004].

3 Cross-Ontology Semantic Web Service Composition

Usually Web service input and output parameters are described by means of XML Schema types which are defined in domain-specific XML standards. But with regard to the composition of services the use of XML Schema yields to significant obstacles. As it has been argued in [Kl2000], the static type bindings do not allow for polymorphism; in particular XML Schema lacks inheritance relations to be exploitable for matching. Therefore as the first step of our approach the domain specific schemas are lifted to an ontology level. Instead of describing input and output parameters of Web services by means of XML Schema types the parameters are being described by concepts which have been defined in domain-specific ontologies. These ontologies are backed with formal logics thus enabling applications to infer facts which have not been explicitly stated.

Combining domain-specific standards with upper ontologies for Web services, such as OWL-S [OWLS], and WSMO [WSMO] Semantic Web services can be wrapped around already existing Web services.

In order to handle the variety of data representations reflecting different granularity requirements semantic interoperability can be achieved by using the concept of semantic bridges [Ma2002]. Semantic bridges are used to describe the relations between distinct concepts defined in different ontologies which are not shared but which intuitionally have an equal or similar meaning. Furthermore semantic bridges define mappings, i.e. a translation between these concepts. However, such transformations can not be expressed directly in common ontology languages like OWL. Therefore we use a rule language for defining semantic bridges. Such a declarative approach has two main advantages. The absence of technical transformation code increases maintainability of the bridges. Furthermore the use of a rule language does not only allow for describing relationships but (given a suitable inference engine) also for performing the transformations between related concepts.

In the following, the approach is illustrated based on the example of two OWL class definitions (`Address` and `PostalAddress`) which, although representing the same concept, have been defined independently in separate ontologies:

[4] This work is supported by the European Commission through the IST-1-002104-STP SATINE (Semantic-based Interoperability Infrastructure for Integrating Web Service Platforms to Peer-to-Peer-Networks) project

```
<owl:Class rdf:ID="Address">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty>
            <owl:Property rdf:ID="hasStreet"/>
        </owl:onProperty>
      <owl:cardinality
        ...
            <owl: Property rdf:ID="hasStreetNumber"/>
        ...
            <owl: Property rdf:ID="hasName"/>
        ...
</owl:Class>
<owl:Class rdf:ID="Name">
        ...
            <owl: Property rdf:ID="hasGivenName"/>
        ...
            <owl: Property rdf:ID="hasSurname"/>
        ...
</owl:Class>
-----------------------------------------------------
<owl:Class rdf:ID="PostalAddress">
        ...
            <owl:Property rdf:ID="hasStreetAddress"/>
        ...
            <owl:Property rdf:ID="hasRecipient"/>
        ...
</owl:Class>
```

The semantic bridge is illustrated in figure 2. It uses rules for mediating between these two representations of an address. Domain standard S1 (e.g. a standard for citizen registration) defines the address attributes with a finer granularity than S2 (e.g. a domain standard for building application) does. Applying the semantic bridge rule shown in figure 2 an instance of type `Address` is furnished with additional properties e.g. with `hasStreetAddress` combining the values of the `Address` properties `hasStreet` and `hasStreetNumber` by means of string concatenation. Likewise `hasRecipient` is constructed by concatenating the properties `hasSurname` and `hasGivenName` from `Name`.

Having the class definitions on hand an OWL-DL reasoner is now able to classify the instance as being a member of the defined class `PostalAddress` since all required properties (including hasStreetAddress) are present. Thus, in the scope of a process any service requiring a `PostalAddress` can now use this instance as it is polymorph of type `Address` and `PostalAddress`. The concept of defined classes in OWL follows the concept of facet classification, i.e. a class is defined in terms of its properties. Any individual featuring such a specific set of properties is then classified as an instance of the class.

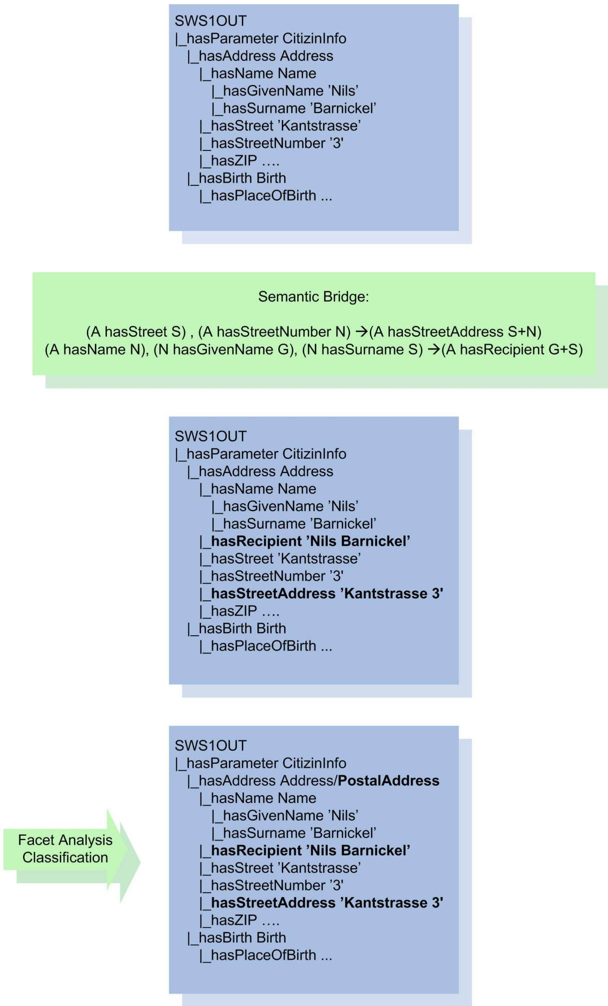

Fig. 2. Semantic Bridging

Using rules for describing semantic bridges enables expressive mappings including one-to-one, one-to-many, and many-to-many. A significant benefit of the presented approach lies in its efficiency: Semantic integration of specific standards has only to be done once on a domain level rather than repeatedly for any service composition, i.e. on the application level. When applying the approach in combination with existing ontology mapping tools [Bi2005,ES2004] to semi-automatically define the semantic bridge rules manual integration efforts can be reduced substantially.

The concept of cross-ontology Semantic Web service composition is illustrated in figure 3. For each domain, experts define a domain ontology besides existing domain XML Schema standards. These domain ontologies are utilized to describe the input and output parameters of Semantic Web services which are wrapped around WSDL-based Web services with XML Schema described parameters. In order to define semantic bridges between different domain ontologies the various experts need to share their knowledge (right hand side of figure 3).

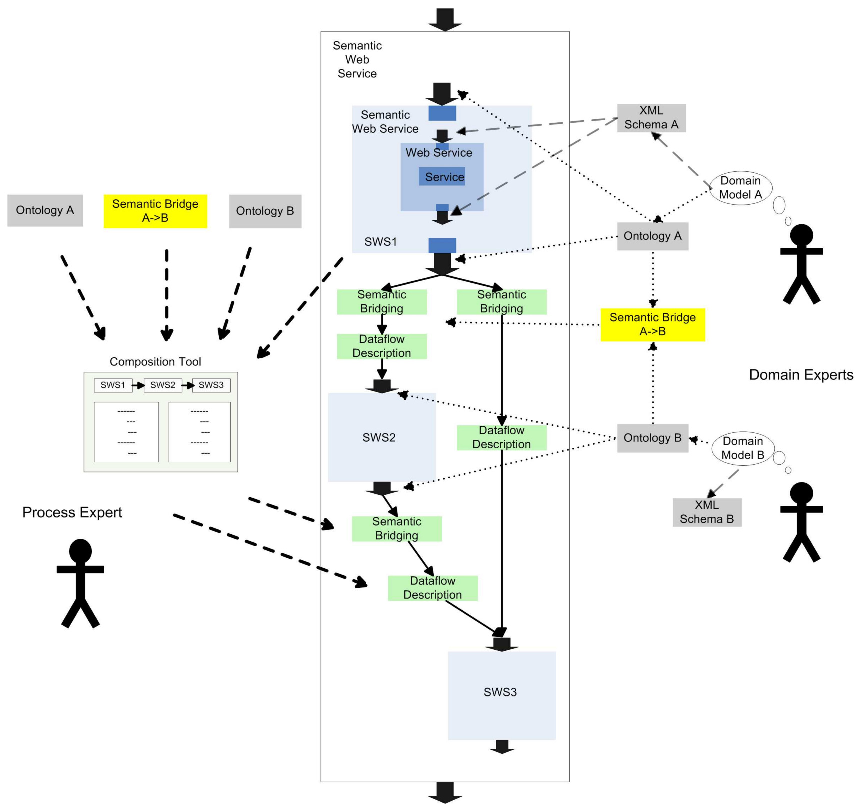

Fig. 3. Cross-Ontology Semantic Web Service Composition

Having established the basis for semantic interoperability semi-automatic tool support during the composition process is given in the following way (left hand side of figure 3): A matching engine performs the reasoning over semantically described relationships (such as inheritance or equality between concepts), thus enabling the composition tool to make recommendations for suitable assignments between output and input parameters of different services. In the background the composition tool applies the semantic bridges. Consequently, for a user different data representations become transparent.

Many matching approaches, such as the one described in [Pa2002], require concepts to be derived from the same ancestors or to be explicitly declared as semantically equal in order to be matched. Applying the semantic bridges helps to overcome this weakness by normalizing the different representations of Web service parameters. After the normalization process conventional matching mechanisms can be utilized.

4 Implementation Issues

In order to validate the presented approach a composition tool has been developed. In our prototypical implementation we have focused on composition support for the design of data flow in service sequences. The eGovernment Web services are wrapped by Semantic Web services, i.e. OWL-S service descriptions have been developed on top of the existing WSDL-based descriptions. Several domain-specific XML standards have been modeled as ontologies which are used to describe the concepts of input and output parameters. In conjunction with the predefined semantic bridges these service descriptions represent the input for our composition tool. The implementation includes apart from the composer, a matching engine, a deployment engine, and an execution engine for the coordination of the composite process. For processing the OWL-S descriptions and for service invocation the Jena framework and the Mindswap OWL-S API were used. The semantic bridges are modeled by Jena rules and are processed using the Jena framework and the Pellet OWL-DL reasoner to implement the matching during design time and mediation at execution time. The dataflow as designed by the user is also described by means of Jena rules. In both cases rules can be implemented as forward chaining rules where triples in the rule head are inferred if the body matches directly. Thus no backtracking is needed and complexity is limited to checking all bodies iteratively until no more rules are fired. Within the rules built-in procedural primitives are used in order to transform different data representations. These built-ins may be extended easily thus allowing for expressive transformations including the creation of new individuals. As future work we are planning to integrate Semantic Web services into the rules (by means of built-in primitives), hence allowing for transformations which can only be made by using external information, e.g. transforming a zip code to a postal district. The output of the composition tool are a proprietary XML-based process execution plan, rule-based dataflow descriptions, and an OWL-S description of the composite service. The OWL-S description is grounded by means of a generic WSDL Web service which acts as an interface to the execution engine.

Figure 4 shows a screenshot of the prototypical composition tool captured during the matching process. The semantic bridge which is applied to enable the matching corresponds to the example presented in 2. As it can be seen in the screenshot, the `hasAddress` property of the SWS1 output is classified as being an instance of both classes, `PostalAddress` and `Address`. Thus, despite the heterogeneity of the underlying data models it is identified as a potential input for SWS2.

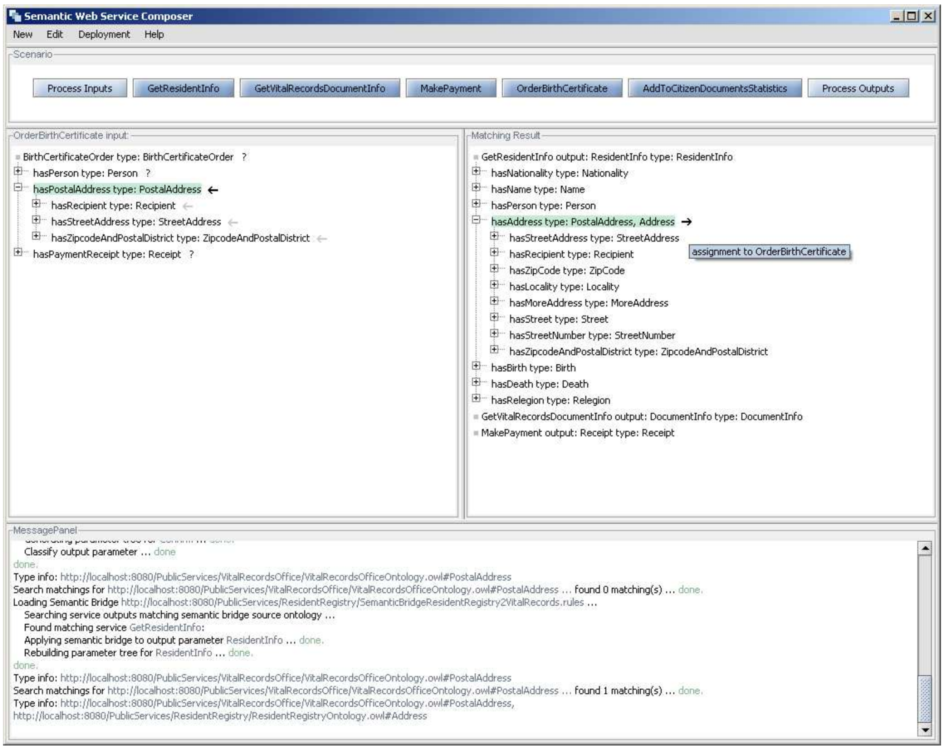

Fig. 4. Semantic Web Service Composer

5 Conclusion and future work

We have presented an approach for supporting the process-oriented integration of heterogeneous eGovernment Web services. During process modeling any transformations between different data representations is done transparently for the user. This is being achieved by semantically bridging independently developed domain standards by means of logical rules which describe the relation and the transformation of concepts from different ontologies.

We think that the presented concepts have several benefits. Firstly, the semantic integration is shifted from the application level to the domain level, thus it has to be done just once rather than for each single service composition. Furthermore the absence of technical transformation code eases the maintainability of the semantic bridges.

By using a rule language for representing the semantic bridges not only the concept relationships can be described but also inference engines can use the rules in order to perform the necessary concept transformations at composition design time and at composition execution time.

The concept of semantic bridges has been used in several approaches for Semantic Web service integration[5]. These approaches implement semantic bridges as separate transformation components (e.g. as external services). Hence the semantic bridge information can not be directly integrated into the inference process. In our approach an inference engine while reasoning over input output parameters directly makes use of the rules that constitute a semantic bridge.

So far, in order to achieve a proof of concept we have just focused on sequences of services. To describe more complex processes additional control flow structures need to be supported. In order to be able to use robust and mature execution engines a mapping from our proprietary process execution plan to the widely-used Business Process Execution Language (BPEL) is planned as future work.

The main challenge in this respect is to find a suitable mapping between different abstraction levels: While at design time ontologies and rules are used for data representation and mediation BPEL execution engines make use of XML Schema types and XSLT transformations.

[5] mediator services in WSMO [WSMO], translator services in Mindswap composer [EB2004]

References

[Bi2005] Veli Bicer et. al.: Providing Semantic Interoperability in the Healthcare Domain through Ontology Mapping, Sigmod Record, Volume 34, Number 3, 2005

[BN2003] Mikkel Hippe Brun, Brian Nielsen: Naming and Design Rules for eGovernment - The Danish Approach, http://xml.coverpages.org/NDR-DanishXML2003.pdf, 2003

[EB2004] Evren Sirin, Bijan Parsia, James Hendler: Composition-driven Filtering and Selection of Semantic Web Services, In AAAI Spring Symposium on Semantic Web Services, 2004

[ES2004] M. Ehrig and S. Staab: QOM - quick ontology mapping. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, Proceedings of the Third International Semantic Web Conference, 2004

[FST2006] Matthias Fluegge, Ivo J. Garcia dos Santos, Neil Paiva Tizzo: Challenges and Techniques on the Road to Dynamically Compose Web Services, to be presented at the Sixth International Conference on Web Engineering ICWE 2006, Palo Alto, California, July 2006

[FT2004] Matthias Fluegge and Diana Tourtchaninova: Ontology-derived Activity Components for Composing Travel Web Services. International Workshop on Semantic Web Technologies in Electronic Business SWEB2004, Berlin, Germany, October 2004.

[Kl2000] Michael Klein et al.: The Relation Between Ontologies and Schema-Languages, Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods, 2000

[Ma2002] A. Maedche et. al.: Mafra - a mapping framework for distributed ontologies. In: Proceedings of the EKAW (Knowledge Engineering and Knowledge Management) 2002, Volume 2473 of Lecture Notes in Computer Science, Springer, 2002

[Mg2006] Gregories Mentzas: Conference Introduction, The Semantic Web meets eGovernment AAAI Spring Symposium Series, Stanford University, 2006

[OWLS] OWL Web Ontology Language for Services (OWL-S), Submission request to W3C, July 2004, www.daml.org/services/owl-s

[Pa2002] Massimo Paolucci et. al.: Semantic Matching of Web Services Capabilities. In: International Semantic Web Conference, pages 333-347, 2002

[WSMO] Web Service Modeling Ontology (WSMO), Submission request to W3C, June 2005, www.wsmo.org

Integration of eGov services: back-office versus front-office integration

Marian Mach[1], Tomas Sabol[2], Jan Paralic[1]

[1]Dept. of Cybernetics and Artificial intelligence, Technical University of Kosice, Letna 9, 042 00 Kosice, Slovakia
[2]Faculty. of Economics, Technical University of Kosice, Letna 9, 042 00 Kosice, Slovakia
{Marian.Mach, Tomas.Sabol, Jan.Paralic}@tuke.sk

Abstract. The paper presents a front-office integration approach applied within the IST Project FP6-2004-27020 Access-eGov. It describes a process how a scenario, corresponding to a particular life-event or business episode of a user, is generated using a front-office integration. The approach proposed provides also additional benefits to the user, e.g. guidance of the user through the scenario by a virtual personal assistant. As a kind of position paper, it argues that the front- and back-office integration are not mutually exclusive approaches, but can also be used in a complementary manner.

1 Introduction

The terms like accessibility, reducing administrative burden on citizens and enterprises, inclusive services, trusted access to eGov services, interoperability etc. currently occur in many reports, papers and presentations. They echoed also in presentations of many speakers and demonstrations of cases submitted to the eEurope Awards for eGovernment - 2005 [1] and were presented by the finalists at the eGovernment Conference. Especially the issues of the interoperability call for special attention. The new Commission's Communication [2] calls for interoperability among all national and regional administrations in the EU

The Commission's Communication has identified basically a need for interoperability at three different levels [3]:

- Interoperability of administrative processes (called organisational interoperability) for:
  - "life-time events" for citizens – e.g. birth, marriage, social security, etc.;
  - "business events" – e.g. setting up a company, paying taxes, participating in procurement activities, etc.;
- Understanding each other's information (semantic interoperability). The systems must "understand" the precise meaning of exchanged information. For example, birth certificates are rather standardised documents but they can look quite differently in different countries.

- Technical interoperability: Computers must be able to “talk” to each other. This is the interoperability level dealing with linking-up systems and is normally tackled via open interfaces, standardisation, access and communication protocols, etc.

This paper explains our approach to the issues mentioned above. The IST Project FP6-2004-27020 Access-eGov [4] is based on a novel approach to the eGovernment service integration - on the users’ side. This approach enables to bridge the gap between existing electronic and traditional services provided by (one or several) public administration(s). These services often need to be combined in order to satisfy the needs of the user (citizen or business) in given situation. The idea of one-stop government related to typical life-event situations [5] is related to the first level of interoperability (i.e. organisational interoperability). On the other hand, semantic web technologies are more targeting semantic and technical interoperability (i.e. the second and third level). The Access-eGov Project is trying to approach all the three levels in an innovative way.

2 Integration of eGov Services

It is not unusual even today that users (either citizens or businesses) are sometimes facing a quite trivial problem – which public administration institution is providing a service(s) they need in the given situation (context), and subsequently to check whether these services are provided in an electronic way or only in a ‘traditional’ way, what inputs are required to this service etc.

This is however only the first (and rather simple) issue. In real life situations citizens, as well as businesses, usually do not need an atomic (singular) government service, but more often a (non-linear) sequence (including *if-then-else* branches). It means a ‘scenario’ of atomic services. And since we are still far away (especially in EU10) from the situation that all the needed government services for the given life event are available on-line, the users usually have to deal with a combination of traditional services and e-services – it means they have to deal with ‘hybrid scenarios’. Of course, carrying out a sequence of (mutually dependent) e-services needs some kind of integration (and this is also still far away from everyday reality). But in fact the user is interested in the final result, and not in the way how it is implemented.

Current approaches to eGov services integration are mostly related to back-office integration. The reorganisation of back offices represents a “hot topic” in current (and not only) European research and is becoming ‘overcrowded’. This applies also to the eGov Project proposals submitted within IST calls of FP6 (e.g. IST-2002-507749 Terregov, IST-2003-507217 eMayor).

3 Front-office integration

Front-office integration can be understood as a synonym for an integration of services on the user (application) level. Our approach to the service integration takes the position not to be invasive to the existing solutions. This enables to index not only services currently without any semantic information attached, but also services developed with semantics in mind – so that they already have a semantic description. This enables to consider not only services which are accessible electronically but also services being delivered in a 'traditional' way (i.e. not via an electronic channel).

To compare back-office and front-office ways of integration of services (and no integration as well), Fig. 1, Fig. and Fig. 3 show high level models of these integration methods.

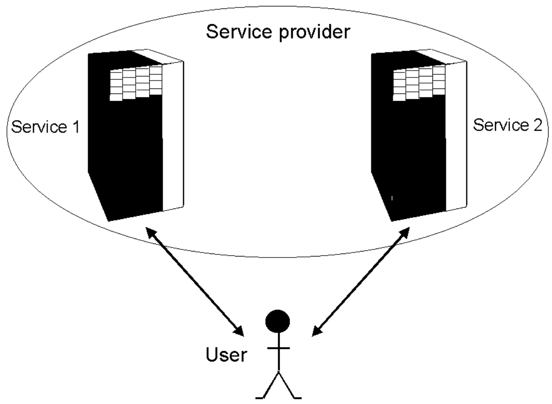

Fig. 1. No integration of services. User accesses each service in an order given by the particular case to be solved.

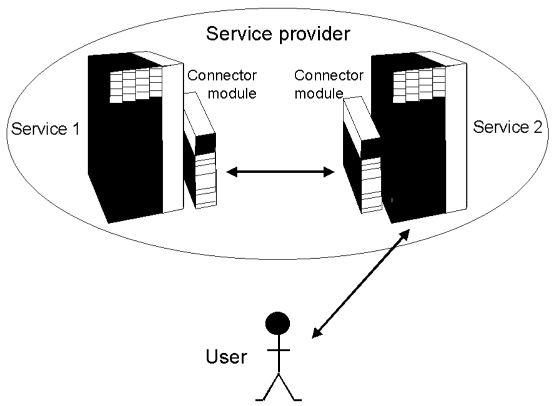

Fig. 2. Back-office integration of services. User accesses one (root) service only. If another service is required to be invoked, the process is performed by the service already in action. Integration is done on system level

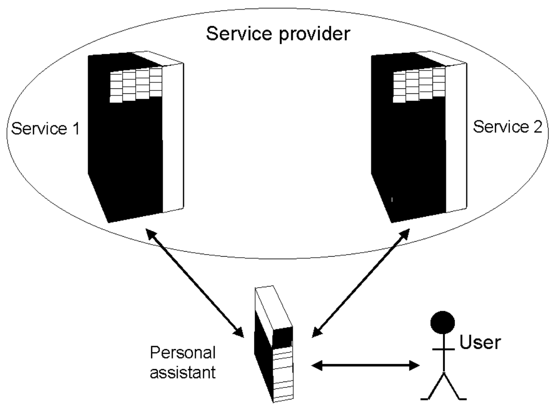

Fig. 3. Front-office integration of services. User accesses only personal assistant. Since services are not integrated, the assistant mediates access to particular services in a proper order. Integration of services is done on user (application) level.

The vision of the Access-eGov Project is to develop component-based enhancements to the existing e-Government infrastructure based on Semantic Web technologies and distributed architectures (Service-oriented and Peer-to-Peer). Service integration is targeted by a 'service composition' layer [6].

The aim of the service composition layer is to generate a complex plan (process) how to cope with a life event (or business episode). In order to satisfy a particular event, a plan (defining: which services should be used, in what sequence, in what way etc.) is expected to be generated. Basically it can be generated in two ways: using top-down or bottom-up approach. The former involves identification of an appropriate generic process definition for a particular life event, its subsequent pruning, and instantiation of particular services.

A life event (or a business episode) is represented as a general scenario for solving the event. The life event scenario is a general one - and will be adjusted/instantiated to a particular user. In order to adjust it to the needs of a particular user, the following steps will be taken:

- To adjust a general event scenario to a particular user in order to specify which steps must be performed;
- To select a service from those being offered for each step in the generated user-specific scenario (e.g. based on the user address, birth place, age etc.);
- To generate a plan for the user how to solve his/her life event.

The plan generated for a particular event represents a guide for user, which can be executed. And since not all the users feel comfortable when dealing with a myriad of public administration services, a virtual personal assistant will guide the user through the generated scenario. The virtual personal assistant will actively help the user to act in accordance with the plan. The role of the personal assistant is to execute a process instance (activity by activity).

Some activities are performed electronically by the assistant (it accesses some eGov services electronically on behalf of the user – the assistant invokes a web service of one institution and translates the response appropriately to supply it as an input to another institution). And some activities (those which are not delivered electronically) must be performed by the user himself. The user is informed on the progress made (mapped to the process activities). Additionally for some activities, the user is asked to carry out some tasks, e.g. to take a form (delivered by the assistant), print, fill, sign, send it by post, go to some office, etc.

The personal assistant represents a temporary 'one-stop shop' dedicated to the particular user and the particular life event and is generated in a dynamic way. Thus, it plays the role of a service integrator – it integrates available services in a "front-office" space. Allocation on the user level enables to integrate traditional services (not accessible electronically) as well.

The advantages of the front-office integration are mainly:

- Integration of services without any intervention into their implementation (possibility of easy integration of legacy systems currently in use; possibility to integrate also 'traditional', non-electronic services);
- Simplicity (there is no need for full integration of services if only some parts of the services are required on the application level – thus saving resources);
- Enriching the integration itself by additional services for deployment of the integration (the integration is performed at application 'foreground' – it is possible to enrich the original services);
- Flexibility (possibility to integrate existing services in different ways depending on the context of the integration; possibility to customise the integration for a particular user and a particular task).

Both the front-office and back-office integration aim at increasing service quality for the end users. The front-office integration can have preference when the existing system is not in crisis and the press for change is manageable within the existing framework [7]. The front-office integration approach can prolong a life cycle of the existing technology solution and thus create a time reserve for better preparation of the back-office integration, which most probably will be necessary at the end of the day (but which is always difficult and costly).

The Access-eGov project started in January 2006, its expected duration is 36 months. The Project is currently in the phase of specifying user requirements. Therefore the idea of implementation of front-office integration of eGov services exists at a generic level only. In the next phase overall system architecture will be designed, including specification of implementing the front-office integration.

4 Conclusions

Both kinds of integration – back-office as well as front-office – are based on the same principles (semantic description of available services). It would be a misunderstanding to understand these two kinds of integration as exclusive "opponents". These approaches can complement each other, to create a synergy increasing benefits for end users. We believe that integration of eGov services in the future will be using a combination of both the approaches, thus transforming them into two fundamental integration pillars.

Acknowledgements

The work presented in the paper was supported by the EC within the FP6-2004-27020 Project "Access to e-Government Services Employing Semantic Technologies" as well as by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/1060/04 Project "Document classification and annotation for the Semantic web".

References

1. eEurope Awards, available at http://www.e-europeawards.org/
2. Commission Communication on interoperability, available at http://europa.eu.int/idabc/en/document/5316
3. European Interoperability Framework for pan-European eGovernment Services, version 1.0, available at http://europa.eu.int/idabc/en/document/3761
4. Van Isacker, K. – Sabol, T.: Project Presentation. Deliverable D9.4, Project FP6-2004-27020 "Access-eGov: Access to eGovernment Services Employing Semantic Technologies", available at www.access-egov.org.
5. Tambouris, E. – Wimmer, M.: On line one stop government: a single point of access to public. In: Digital Government: Startegies and Implementations in Developed and Developing Countries, Wayne Huang (ed.), Idea Publishing Group, 2004.
6. Mach, M. – Sabol, T. – Paralic, J.: Improving Access and Efficient Use of Government Services by Means of Semantic Technologies. Submitted to the 5th Int. EGOV Conference 2006 within the DEXA conference cluster , Krakow, Poland.
7. Millard, J. et al.: Reorganisation of Government Back Offices for Better Electronic Public Services – European Good Practices (Back-office reorganisation) Final Report to the European Commission, January 2004, available at http://www.epsigate.org/sample_content/back_office_reorganisation_final_report.pdf .

A Three Dimensional Framework to Realize Interoperability in Public Administrations

Elisabeth Diedrich, Daniel Schmidt, and Maria Wimmer

University of Koblenz-Landau, Institute for IS Research
Universitaetsstr. 1, 56070 Koblenz, Germany
{lisadied|dmschmidt|wimmer}@uni-koblenz.de

Abstract. A major challenge for European Governments is to improve ways of heterogeneous systems working together. Due to the growing of the European Union, existing systems have to be analyzed and evaluated properly to pave the way for a smooth cooperation and collaboration. This paper presents a framework as a holistic approach in providing (developing) a three-dimensional model for interoperability in the large. After motivating the need for such a framework and presenting the EC's activities in this sector, section 2 exemplifies interoperability in todays eGovernment. This leads to the introduction of the R4eGov interoperability framework and the methodology to vitalize this framework, before we close with the conclusion.

Keywords. interoperability, framework, eGovernment

1 Motivation and Prerequisites

The European Union keeps growing and member states become more cross-linked every day. Some reasons are that governments are requested to work together more frequently, more intensely and in a vast and ever evolving environment. The drivers of change are manifold: modernization, a huge gap between the burden of work and the available resources, new legal settings and strategic commitments, new ICT[1], keeping up with the change taking place in private business settings, higher expectations for improved quality of service, enhanced public value generation, etc. One could list a large number of aspects implying cooperation among public administrations and cooperation with their stakeholders on the basis and by means of advanced ICT.

To enable cooperation (either in terms of collaboration or coordination), two approaches can be identified: integration and interoperation. Klischewski and Scholl define integration as "the forming of a (temporary or permanent) larger unit of government entities for the purpose of merging processes [and systems,] and/or sharing information" [1]. The European Commission has defined interoperability as "the means by which the inter-linking of systems, information and ways of working, whether within or between administrations, nationally or across

[1] Information and Communication Technologies

Europe, or with the enterprise sector, occurs" [2]. Klischewski and Scholl further stress that systems and applications that interoperate are characterized by the following aspects: independency, heterogeneity, and control by different jurisdictions/administrations or by external actors; yet also cooperation in a predefined and agreed upon fashion [1]. Likewise, Wimmer et al stress that interoperation can only be reached by means of open standards [3], whereby interoperation needs to be addressed on technical, semantic and organizational level alike (cf. [2] and [3]).

As can be recognized, interoperability is the means to pave the way to smooth cooperation. In order to improve interoperation, the European Commission has launched several programs to fund research and development in interoperability. Examples are the Interchange of Data between Administrations (IDA, within which the European Interoperability Framework (EIF) has been developed) and the Interchange of Data between Administrations, Businesses and Citizens (IDABC) [4] or the MODINIS interoperability framework [5].

In spite of the different approaches mentioned above the framework presented in this paper shows an holistic approach to get a 3-D model of the interoperability landscape. The EIF proposes an interoperability view on the three dimensions technical, semantical and organizational. Our approach shifts these three dimensions to layers on one dimension, as further described in section 3.

The need for improved cooperation among administrations themselves and with their external stakeholders (citizens, companies, other non-profit organizations) has also led to a series of research and development projects funded by the EC (to name a few of these projects[2]: ATHENA, BRITE, GUIDE, INTELCITIES, ONTOGOV, QUALEG, R4eGov, SemanticGov, SMARTGOV, TERREGOV). These projects shall put forward solutions for interoperability in various settings of public sector activity.

Within this contribution, we refer to R4eGov[3], an integrated project started under IST, call 4, of the EC in March 2006. In the next sections, we introduce the interoperability framework and approach of R4eGov. We start with a scenario for interoperability by large, thereby sketching the breadth and depth of interoperability. Section 3 describes the interoperability framework and approach suggested for R4eGov.

2 Scenario for Interoperability by Large

Providing an eGovernment framework for interoperability by large requires to address external and internal system components to interoperate alike. A typical scenario of interoperation among internal and external system components is depicted in figure 1. As shown in the figure, an external stakeholder may access a service through a local portal or even through a one-stop government portal that routes the application to a certain authority. In both cases, the portals may use online forms or other ways of interaction to collect the customer's request. At

[2] see `http://ec.europa.eu/research/fp6/index_en.cfm`
[3] `http://www.r4egov.info`

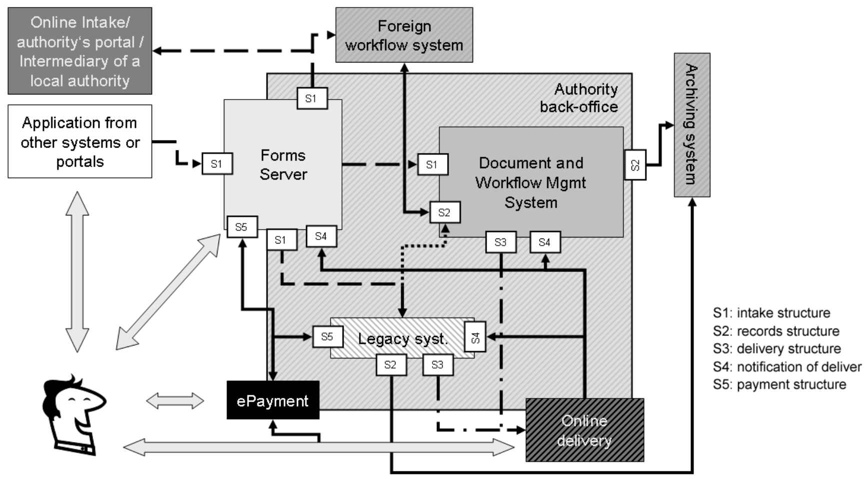

Fig. 1. Scenario of applications and interfaces interacting throughout service performance

the level of data, a standardized format may be established and agreed among the external one-stop portal and the local online service intake counter (S1). This format rules how data concerning the application are collected. Finally, also a common protocol needs to be agreed to ensure smooth interoperation among these two components. In the next phase of process, the intake counter (e.g. a forms server) forwards the application either to an internal workflow management system, a specific domain application or even to another external workflow system or portal (because the invoked service is being provided by some other organization). In any case, interoperation needs to be secured through commonly agreed standards on protocol, data and process level (exemplified with S2). Likewise, the figure indicate further needs for interoperability among systems and components (demonstrated with S3 - S5).

Figure 1 is an exemplary scenario of a public administration's systems and components landscape, whereby the components may be owned and controlled by different authorities or even private service offerers. The components may be spread in an open environment of the WWW (such as a payment service, a delivery service or a one-stop portal) or even be secured in certain organizational system environments (indicated in the figure with 'Authority Back-office').

The vision of eAdministration by large is to enable a smooth interaction and throughput of service delivery without any media break and without problems of inconsistent interoperation between the systems, data storages and components being interlinked in order to provide the full online service execution. Consequently, interoperability has to be addressed in a structured and holistic way. Above all, distinct levels of interoperation - technical, semantical and organizational - have to be secured. The next section introduces the interoperability framework as being used in R4eGov.

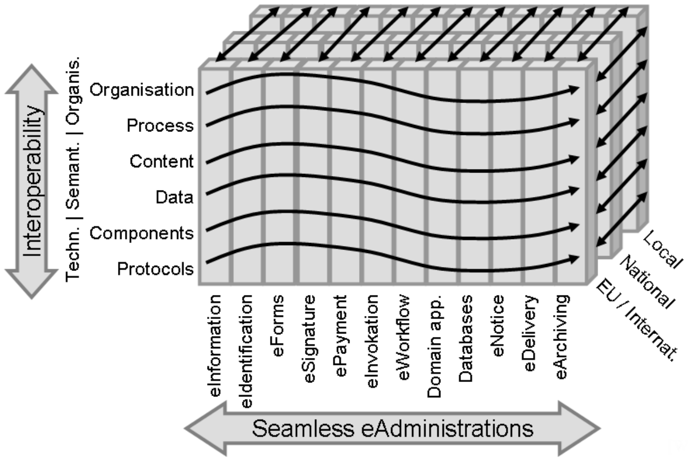

Fig. 2. R4eGov conceptual framework for interoperability by large

3 R4eGov Interoperability Framework

The R4eGov approach to interoperability by large bases on the scenario as depicted in section 2 as well as the requests to address interoperability at technical, semantic and organizational level at the same time. Figure 2 demonstrates the approach, which is based on [3]. We define the three dimensions of this framework as follows.

- *Interoperability:* Making collaboration possible from the technical basis with the semantical knowledge to the organizational setup.
- *Seamless eAdministration:* eAdministration is not represented by one process or service. Actually, most diverse services and processes combine to the big picture of eAdministration. In a collaborative environment, each part of eAdministration has to work smoothly in this compound structure.
- *Organizational hierarchy on state level:* The consolidation of the European Union is still ongoing. The majority of eGovernment systems will remain heterogeneous for the next years as setup and definition of processes will stay under the responsibility of local public administrations. eGovernment applications have to be adapted to local, national or EU/international characteristics.

The framework shall guide system designers to develop interfaces and open standards that guarantee a smooth execution of public services covering the whole life cycle of service execution from intake till archiving. At the same time, the framework shall support to define interfaces for global usage by applying open standard protocols, by developing common data specifications, common process models and by commonly agree on policies to interact in smooth service provision across distinct organizational settings.

R4eGov's interoperability by large, based on the above interoperability framework, will be reached with the following methodology, composed of a state-of-play analysis, requirements analysis, design of a tool to overview and manage interoperability and the development of a life cycle methodology to manage interoperability.

3.1 State-of-play analysis

There are combined and unified systems already in place. There will be additional systems prepared for interoperation in the future. Since a lot of interoperability approaches already exist or are being developed, an investigation shall elicit, which specifications, approaches and solutions are available. This investigation will examine the various levels of interoperability and the distinct phases of process execution at local/regional, national and EC/international level of governments. Special attention will be directed on

- methodological approaches;
- approaches to organizational, semantical and technical interoperability;
- the analysis of trendy concepts like SOA[4], MDA[5], EAI[6]/GAI[7] or Shared Service Centers[8];
- solutions and tool support to manage interoperability at the various levels of interaction in collaborative, cross-organizational workflows and networked governments.

3.2 Requirements derived from real usecase environments

The investigations will be guided by several usecase applications from distinct sectoral areas of eGovernment applications selected for the R4eGov project. The interoperability needs shall be described according to the scenario shown in 1. The usecases will support the work on different levels:

- Analyze and study the critical aspects of interoperability for eAdministration by large.
- Analyze needs and collect IOP requirements of European administrations.
- Collect requirements for the supporting methodology and tools to manage interoperability.

3.3 A tool to overview and manage interoperability

In order to support the management and overview of interoperability developments, a web-based tool shall help to structure, classify and relate specifications, agreements and definitions that contribute to interoperability by large.

[4] Service Oriented Architecture
[5] Model Driven Architecture
[6] Enterprise Application Integration
[7] Government Application Integration
[8] Common GxC/GxB service center providing services of all public administrations

3.4 A life cycle methodology to manage interoperability

In order to support the maintenance and further developments of interoperability, an engineering and maintenance life cycle methodology specifically targeted for guaranteeing interoperability by large in networked administrations shall be defined.

The framework and approach introduced above shall help to overcome current weaknesses and problems of single isolated solutions. It shall provide a means to set up an environment of sharing open standards that will allow new architectural solutions to be used and enable networked, better government to be implemented. With the instruments described, the interoperability landscape shall be investigated and visualized at local, national and EU/international eGovernment contexts.

Acknowledgements. R4eGov is funded by the EU under IST contract no. 26650.

References

1. Klischewski, R., Scholl, H.J.: Information quality as a common ground for key players in e-government integration and interoperability. In: Proceedings of HICSS'06. (2006)
2. EC: Commission of the european communities. linking-up europe: The importance of interoperability for e-government services. Staff Working Document (2003)
3. Wimmer, M., Liehmann, M., Martin, B.: Offene standards und abgestimmte spezifikationen - das osterreichische interoperabilittskonzept. Proceedings MKWI (2006)
4. EC: Commission of the european communities. european interoperability framework for pan-european egovernment services. http://europa.eu.int/idabc/en/document/3761 (2004)
5. MODINIS: Modinis programme. study on interoperability at local and regional level. 2006. (http://www.egov-iop.ifib.de/index.html)