BENEFITS AND CHALLENGES OF APPLYING SEMANTIC WEB SERVICES IN THE E-GOVERNMENT DOMAIN

Alessio Gugliotta[1], Vlad Tanasescu[1], John Domingue[1],
Rob Davies[2], Leticia Gutiérrez-Villarías[2], Mary Rowlatt[2],
Marc Richardson[3], Sandra Stinčić[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 Group
Adastral Park Martlesham, Ipswich IP5 3RE, UK
{marc.richardson, sandra.stincic}@bt.com

Abstract

Joining up services in e-Government usually implies governmental agencies acting in concert without a central control regime. This requires the sharing of scattered and heterogeneous data. Semantic Web Service (SWS) technology can help to integrate, mediate and reason between these datasets. However, since few real-world applications have been developed, it is still unclear which are the actual benefits and issues of adopting such a technology in the e-Government domain. In this paper, we contribute to raising awareness of the potential benefits in the e-Government community by analyzing motivations, requirements, and expected results, before proposing a reusable SWS-based framework. We demonstrate the application of this framework by a compelling use case: a GIS-based emergency planning system. We illustrate the obtained benefits and the key challenges which remain to be addressed. The work described is supported by the DIP (Data, Information and Process Integration with Semantic Web Services) project. DIP (FP6 – 507483), an Integrated Project funded under the European Union's IST programme.

1. Introduction

Currently, one of the main needs of Semantic Web [1] is the development of real-world applications that demonstrate its added (business) values. Thus, the next application-driven research challenge can be defined only through the feedback from practical prototypes and applications. The full potential application of Semantic Web Services (SWS) technology requires many more large-scale testing domains.

Since it is an enormous challenge to achieve interoperability and to address semantic differences related to the great variety of datasets and information technology solutions which should be networked, e-Government may be a very effective test-bed for effectively evaluating SWS frameworks. Moreover, it exhibits further significant characteristics, which may indicate several research issues for SWS: for example, e-Government is characterized by top-down prescribed constraints in key areas (e.g. laws, legal requirements, policies in the use of services and access to data); limited central control; strong requirements to come to similar decisions in similar situations; a high requirement for non-functional properties such as security, privacy, and trust; wide information imbalances between stakeholders, as well as multiple and heterogeneous stakeholders involved in the same process.

The ability to aggregate and reuse diverse information resources relevant to a given situation in a cost-effective way and to make this available as a basis for transparent interaction between community partner organizations and individual citizens is a key benefit that SWS technology can provide to e-Government.

The e-Government community (stakeholders, administrations, end-users, but also researchers) needs to perceive these benefits more clearly before it will adopt the technology. At present, Web Services (WS) are being introduced as infrastructure (often experimental) in some areas of government and the broad awareness of need for semantic enrichment is increasing. However, since SWS are completely new - and are mainly visible to the academic/industrial research 'e-Government' sector - a measurable benefit to service and achievable cost savings, or "cashable benefits" will need to be established. In absence of gold standards, demonstrating real-world applications is an important first step to accomplish this goal.

In the DIP project, e-Government use case, a close collaboration has been established with the Essex County Council (ECC) - a large local authority in South East England (UK) containing a population of 1.3M – to deploy real-world SWS-based applications in the e-Government domain. In this paper, we report our experience, in order to provide a proof of concept of the necessary added value and to outline future research directions.

2. Deploying SWS-based Applications

SWS research aims to automate the development of WS based applications through the Semantic Web technology. By providing formal descriptions with well defined semantics we facilitate the machine interpretation of WS descriptions. The key areas of concern are the following:

- *Discovery:* finding WS which can fulfil a task. Discovery usually involves matching a formal task description against semantic descriptions of Web services.
- *Mediation*: we can not assume that the software components which we find are compatible. Mediation aims to overcome all incompatibilities involved. Typically this means mismatches at the level of data format, message protocol and underlying business processes.
- *Composition*: often no single service will be available to satisfy a request. In this case we need to be able to create a new service by composing existing components. AI planning engines are typically used to compose Web service descriptions from high goals.

In the last years, several projects applied SWS technology in the e-Government domain – [10] and [11] to quote a few examples -, but only a few of them show reusability and composability in real usage scenario. Differently, [8] proposes solutions to a specific real usage scenario by using ad-hoc ontologies to automatically generate WS customized to senior citizen's needs and government program laws and regulations. This highlights the practical applicability of its approach, but limits the reusability and flexibility. No one of the mentioned approaches adopts mediation mechanisms to overcome data and process mismatches: they only propose centralized ontologies for representing the entire domain and, thereby, addressing interoperability.

In our work, we defined an open and flexible approach that allows us to deploy real world applications taking advantage of SWS technology to introduce added value services as well as reflect the e-Government structure.

In order to provide semantics and step toward the creation of added value services, we adopt WSMO (Web Service Modelling Ontology) [12] – a promising SWS framework – and IRS-III – a tested implementation of this standard [2]. WSMO is a formal ontology for describing the various aspects of services 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.

2.1 Application General Architecture

Since government legacy systems are often isolated - i.e. not interconnected and/or use distinct technological solutions -, our approach firstly enables the data and functionalities provided by existing legacy systems from the involved governmental partners to be exposed as WS, which are then semantically annotated and published using our SWS infrastructure. The following architecture reflects and explains this double stage process.

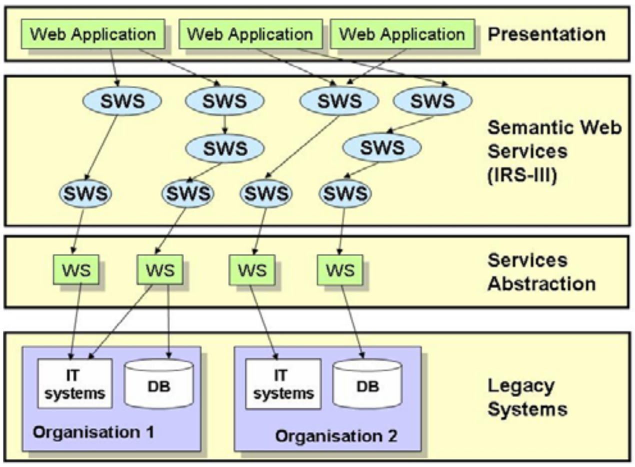

Figure 1. The generic architecture used when creating IRS-III based e-Government applications.

- *Legacy System layer:* consists of the existing data sources and IT systems available from each of the parties involved in the integrated application.
- *Service Abstraction layer:* exposes (micro-) functionality of the legacy systems as WS, abstracting from the hardware and software platforms. In general existing Enterprise Application Integration (EAI) software will facilitate the creation of required WS. Note that for standard databases the necessary functionalities of the WS can simply be implemented as SQL query functions.
- *Semantic Web Service layer*: given a goal request this layer, implemented in IRS-III, will (i) discover a candidate set of Web services, (ii) select the most appropriate, (iii) mediate any mismatches at the data, ontological or business process level, and (iv) invoke the selected Web services whilst adhering to any data, control flow and Web service invocation requirements. To achieve this, IRS-III utilises the set of SWS descriptions, which are composed of goals, mediators, and Web services, supported by relevant ontologies. Note that we distinguish two main sets of SWS descriptions: *basic SWS* (bottom of the layer) that simply wrap the WS to fulfil goals; and *complex SWS* (top of the layer) that require a composition of basic or complex SWS to fulfil complex goals.
- *Presentation layer:* is a Web application accessible through a standard Web browser. The goals defined within the SWS layer are reflected in the structure of the interface and can be invoked either through the IRS-III API or as an HTTP GET request. The goal requests are filled with data provided by the user and sent to the Semantic Web Service layer. We should emphasise

that the presentation layer may be comprised of a set of Web applications to support distinct user communities. In this case, each community would be represented by a set of goals supported by community related ontologies.

As can be seen, the architecture can be compared with well known service oriented architectures. The added value is introduced at the *Semantic Web Service layer* where integration and interoperability of existing heterogeneous services are accomplished at run-time.

2.2 Development Methodology

In order to successfully create applications from SWS, four key activities need to be carried out as follows:

1. *Requirements capture:* the requirements for the overall application are captured using standard software engineering methodologies and tools. We do not advocate any particular requirements capture method but envisage that the resulting documents describe the stakeholders, the main users, roles, and goals, any potential providers for Web services, and any requirements on the deployed infrastructure and interfaces.
2. *Goal description*: using the requirements documents above relevant goals are identified and semantically described in IRS-III. During this process any required supporting domain ontologies will either be created from scratch or existing ontologies will be re-used.
3. *Web service description*: descriptions of relevant Web services are created within the IRS. Again, any domain ontologies required to support the Web service descriptions are either defined or re-used as necessary.
4. *Mediator description*: mismatches between the ontologies used, and mismatches within and between the formal goal and Web service descriptions are identified and appropriate mediators created.

The first two steps are user/client centric and therefore involve discussions with the relevant client stakeholders and domain experts, whereas step 3 will require dialogue with the Web service providers and domain experts. Steps 2 and 3 are mostly independent and in the future we expect libraries of goals and Web services to become generally available to support reuse. Finally, Steps 2, 3, 4 are supported by IRS-III. As a result, we obtain a semi-automatic knowledge acquisition process for the development of our applications.

3. Expected Benefits

In the following we summaries the benefits of our approach over other technologies:

- *SWS vs. Web Services*. Using WS, data and functionalities can be shared with anyone through the Internet. The supplied services are autonomous and platform-independent computational elements. The syntactic definitions used in these specifications allow fast composition and good results in term of application performance. However, they do not completely describe the capability of a service and cannot be understood by software programs. A human developer is required to interpret the meaning of inputs, outputs and applicable constraints, as well as the context in which services can be used. Moreover, WS lack in flexibility; for instance, if a new WS is deployed, the application developers need to re-model several syntax descriptions – introducing a cost - in order to integrate it in a specific context. On the other hand, the SWS approach is able to model the background knowledge of a context together to the requested and provided capabilities, and hence address *automatic reasoning* and *reuse*. In this way, service invocation, discovery, composition, and mediation are *automated* by adopting the best available solutions for a specific request increasing the *flexibility*, *scalability,* and *maintainability* of an application. In our approach, the execution sequence of a complex SWS is not hard-coded, but it is dynamically created using a goal-based discovery and invocation: several WS may be associated with a goal, and only the best one will be discovered and invoked at runtime only; if a new service will be available, the developers simply will describe and then link it to an existing goal; if a service will change, only the specific semantic description will be affected, and not the whole business process. Finally, developing WS takes time. However, our approach allows the publishing of SWS simply from lisp functions and java methods, reducing the time of deploying legacy system functionalities.

- *SWS vs. other ontology-based approaches*. Creating and managing ontologies is a bottleneck: understanding a domain, acquiring and representing knowledge, populating with instances and evolving ontologies are big tasks for the application developers. However, in the context of a semantic-based application, this is a cost that cannot be deleted, but one that may be contained. In complex domain such as e-Government, centralized ontologies would require an unrealistic development effort with no guarantee of satisfactory results in terms of capturing domain knowledge. Moreover, government agencies deal with huge datasets (e.g. demographic, GIS, etc.) that cannot easily transposed to ontology's instances. SWS technology makes *knowledge capture* and *maintenance* process *simpler* and more *efficient*: (i) the only knowledge which must be modeled is related to the exposed functionality implemented by the WS involving simply the concepts used by WS and the instances created (lifted) when invoking the Web service. This minimalist approach also improves the management of the ontology evolution and maintenance; (ii) the knowledge capturing is distributed among all of the stakeholders: each partner describes – and it is responsible for – its particular domain; in this way, the several viewpoints (requesters and providers) can be independently and concurrently described by the proper knowledge holders. Partners can also reuse already existing ontologies. As a result, we obtain a *model that reflects the e-Government structure* (i.e. conforming to the specific situation) and addresses the required *lack of central control*.

Differently from both WS and other semantic approaches, our WSMO based approach can support dealing with *interoperability* among heterogeneous knowledge sources and *mediation* among several viewpoints (users, multiple providers, etc.). WSMO mediators are mappings that solve existing mismatches and do not affect service descriptions. In our applications, we have gathered the following mediation requirements and solutions*:*

- *Data mediation (oo-mediators)*: organizations have their own databases and hence different data formats for the same concept. Different data formats can be lifted to same or multiple concepts within domain ontologies. At a semantic level, different concepts can be mapped through mediators.
- *Goal mediation (wg-mediators)*: organizations can define one goal that can be satisfied in different ways by applicable Web services developed within different agencies. Multiple Web services can be linked to the same goal via mediators.
- *Process mediation (gg-mediators)*: organization processes behave in different ways according to their own set of operational procedures, requirements and constraints. Each Web service presents a choreography describing how a client talks to the deployed service. Furthermore, sub-Goals can be composed together for providing the functionality of one Web service through the orchestration.

4. A real world application: an Emergency Management System

In this section, we demonstrate the applicability of our approach (Section 2) and verify the expected benefits (Section 3) by describing a compelling use case in the e-Government domain: an Emergency Management System (EMS).

In emergency situations, 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 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 spatially-related. Spatially-Related Data (SRD) is traditionally managed with the help of Geographical Information Systems (GIS) which 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.
Following several interviews with SRD holders in ECC, it was decided to focus the scenario on the ECC Emergency Planning department, specifically on a real past emergency situation: a snowstorm which affected the M11 motorway on 31st January 2003. The EMS prototype is in effect a decision support system, which assists the end-user – currently the Emergency Planning Officer (EPO), but extensible to other emergency management personnel: ambulance service, fire service, police, etc. – in assembling information related to a certain type of event, more quickly and accurately.

4.1 Legacy Systems Layer

The EMS aggregates data and functionalities from three different sources:

- *Meteorological Office*: a national UK organization which provides environmental resources and in particular weather forecast data.
- *ViewEssex:* a collaboration between ECC and British Telecommunications (BT) which has created a single corporate spatial data warehouse, containing a wide range of data including roads, administrative boundaries, buildings, and maps, as well as environmental and social care data. Within the application we used building related data to support searches for suitable rest centres.
- *BuddySpace:* an Instant Messaging client facilitating lightweight communication, collaboration, and presence management [3] built on top of the instant messaging protocol Jabber. The BuddySpace client can be accessed on standard PCs, as well as on PDAs and mobile phones.

4.2 Service Abstraction Layer

We distinguish between two classes of services: data and smart. The former refer to the three data sources introduced above, and are exposed by means of Web Services:

- *Meteorological service*: this service provides weather information (e.g. snowfall) over a specific rectangular spatial area.
- *ECC Emergency Planning services*: using the View*Essex* data, each service in this set returns detailed information on a specific type of rest centre within a given circular area.
- *BuddySpace services*: these services allow presence information on online users to be accessed.

Smart services represent specific emergency planning reasoning and operations on the data provided by the data services. They are implemented in a mixture of Common Lisp and OCML [9] and make use of the EMS ontologies. In particular, we created a number of filter services that select the GIS data according to emergency-specific requirements (e.g. rest centres with heating system, hotels with at least 40 beds, easy accessible hospital, etc.). The criteria used were gained from our discussions with Emergency Planning Officers (EPOs).

4.3 Domain Ontologies for the Semantic Web Service Layer

As stated above, the SWS layer is comprised of descriptions within the IRS-III. The goals, mediator and Web Service definitions use ontologies which reflect the client and provider domains. The following ontologies were developed to support the WSMO descriptions.

- *GUI ontology*: composed of user-interface concepts and used to display the results of invocations to the IRS-III. For the EMS application we instantiated the ontology to reflect the Google Maps API.
- *Archetypes ontology* [7]: this ontology provides a cognitively plausible description of geographical objects. It is assumed that any client, whilst maybe lacking the specific knowledge for domain specific concepts, will be familiar with the archetypes contained in this ontology.
- *SGIS spatial ontology*: describes the concepts commonly found in GIS, such as points, spatial objects with attributes, polygons, and fields.
- *Meteorology*, *ECC Emergency Planning* and *Jabber* domain ontologies

4.4 WSMO descriptions for the Semantic Web Service Layer

The goals, mediators, and Web Services descriptions of our application link the UK Meteorological Office, ECC Emergency Planning, and BuddySpace Web services to the user interface. Correspondingly, the Web Service goal descriptions use the SGIS spatial, meteorology, ECC Emergency Planning and Jabber domain ontologies whilst the goal encodings rely on the GUI and archetypes ontologies. Mismatches are resolved by the defined mediators. For illustration purposes, a small portion of the SWS descriptions are shown in Figure 2.

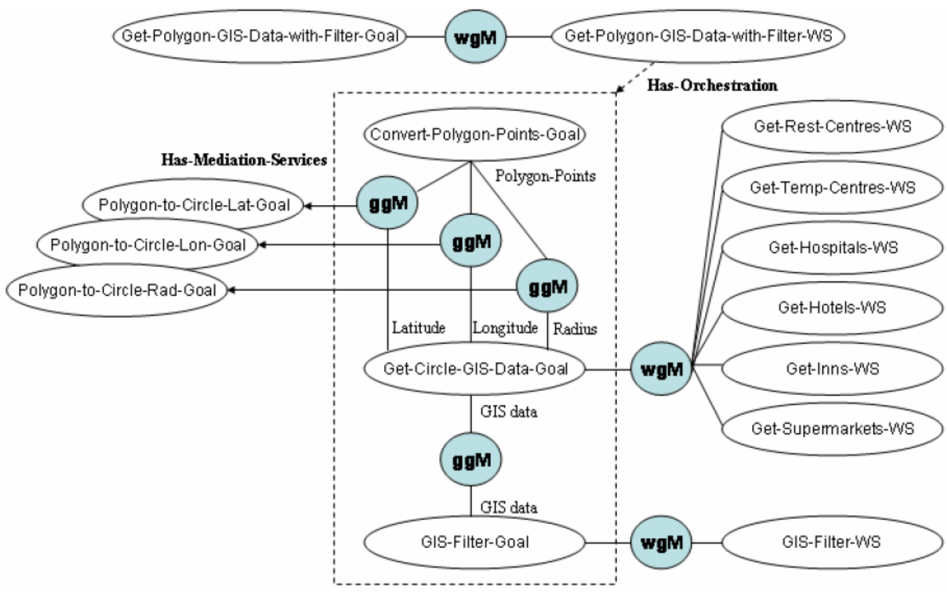

Figure 2. A portion of the WSMO descriptions for the EMS application.

Get-Polygon-GIS-data-with-Filter-Goal represents a request for available shelters within a delimited area. The user specifies the requirements as a target area, a sequence of at least three points (a polygon), and a shelter type (e.g. hospitals, inns, hotels). As mentioned above the set of ECC Emergency Planning Web services each return potential shelters of a specific type with a circular query area. The obtained results need to be filtered in order to return only shelters correlated to emergency-specific requirements (for example a snowstorm). The process automated in our application is usually performed by EPO manually.

From a SWS point of view the problems to be solved by this particular portion of the SWS layer included: (i) *discovering* the appropriate ECC Emergency Planning Web service; (ii) *meditating* the difference in area representations (polygon vs. circular) between the goal and Web services; (iii) *composing* the retrieve and filter data operations. Below we outline how the WSMO representations in Figure 2 address these problems.

- *Web service discovery:* each SWS description of ECC Emergency Planning service defines, in its capability, the specific class of shelter that the service provides. Each definition is linked to the *Get-Circle-GIS-Data-Goal* by means of a unique WG-mediator (shown as wgM). The inputs of the goal specify the class of shelter, and the circular query area. At invocation IRS-III discovers through the WG-mediator all associated Web services, and selects one on the basis of the specific class of shelter described in the Web service capability.
- *Area mediation and orchestration*: the *Get-Polygon-GIS-data-with-Filter-Goal* is associated with a unique Web service that orchestrates, by simply invoking three sub-goals in sequence. The first gets the list of polygon points from the input; the second is *Get-Circle-GIS-Data-Goal* described above; 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 (depicted as ggM) that return the centre, as a latitude and longitude, and radius of the smallest circle which circumscribes the given 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 related WG-mediator and Web service ovals were omitted to avoid cluttering the diagram). The results of the mediation services and the class of shelter required are provided as inputs to the second sub-goal. A unique GG-mediator connects the output of the second to the input of the third sub-goal. In this instance no mediation service is necessary.

It is important to note that if new WS – for instance providing data from further GIS are available, new Web Service descriptions will be simply introduced, and linked to the *Get-Circle-GIS-Goal* by the proper mediators (even reusing the existing ones, if semantic mismatches do not exist), without affecting the existing structure. In the same way, new GIS filter services (e.g. more efficient ones) may be introduced. The effective workflow – i.e. which services are invoked – is known at run-time only.

4.5 Presentation Layer

The user interface has been developed using Web standards: XHTML and CSS are used for presentation, JavaScript (i.e. EcmaScript) is used to handle user interaction and AJAX provides IRS-III goal invocation. One of the main components of the interface is a map, which uses the Google Maps API [5] to display polygons and objects (custom images) at specific coordinates and zoom levels. These objects are displayed in a pop-up window or in a hovering transparent region over the maps.

When the application is launched, a goal is invoked for the Essex region, and snow hazard or storm polygons are drawn according to data from the meteorological office. The value from which snow values can constitute a hazard or a storm are heuristic and as emergency knowledge is gathered it can easily improved, by modifying the smart services which are composed with weather information, while the goal visible to the user remains the same. As an example of practical usage, we describe how an EPO describes and emergency situation, before trying to contact relevant agents. The procedure is as follows:

1. The EPO clicks within the displayed hazard region to bring up a menu of available goals. In this case (Figure 3a) three goals are available: show available shelters, login to BuddySpace and get the presence information for related staff.
2. The EPO asks for the available Rest Centres inside the region, and then inspects the detailed attributes for the Rest Centre returned (Figure 3b).

3. The EPO requests to see the presence status for all staff within the region and then initiates an online discussion the closest online agency worker (Figure 3c).

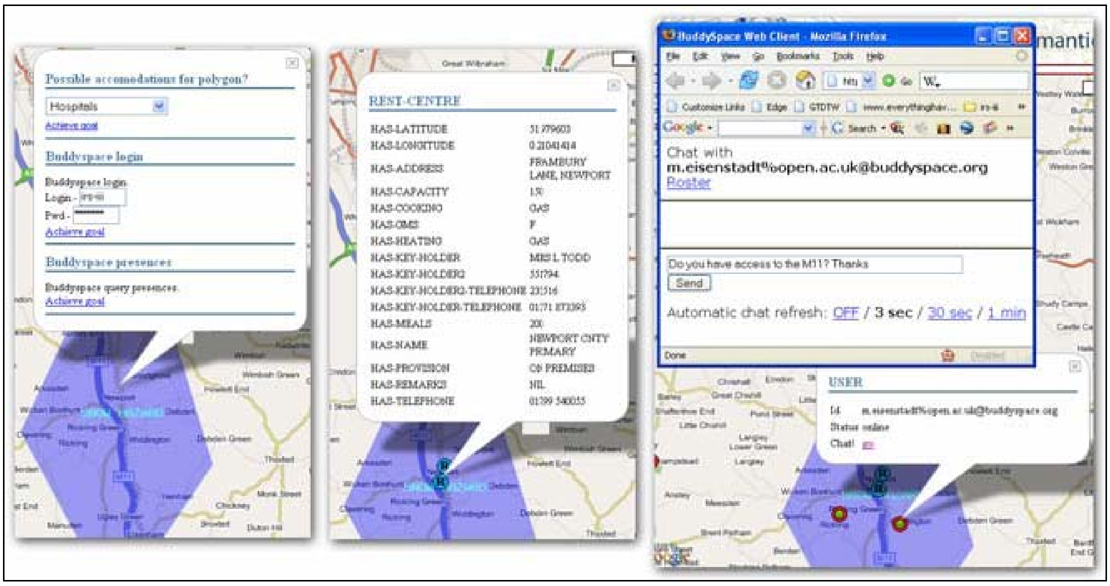

Figure 3 - Three views of the application in use: 3a) Goals available for the snow hazard, 3b) obtaining detailed information for a specific rest centre, 3c) initiating a discussion with an online emergency worker.

4.6 Results

In order to deploy the application, we followed a prototyping approach that produced two main cycles, and a third one is under way. The result of each cycle has been valued by involved stakeholders (emergency planning department in ECC). On the basis of their feedback the subsequent cycle has been planned. The effective involvement of stakeholders in the development process introduced in Section 2.2 eases the adoption of the deployed application within the emergency structures: (i) stakeholder's requirements and knowledge have been captured and used in the application; (ii) existing processes have been represented, eventually simplified, and automated.

The resulted EMS is a decision support system, which assists the EPO in the tasks of retrieving, processing, displaying, and interacting with only emergency relevant information, more quickly and accurately. A screencast of the interaction as well as a live version are available online[1].

Our approach presents many advantages compared to standard based approaches as the one demonstrated in the OWS-3 Initiative[2]:

- *Framework openness*: standards are helpful but not necessary. For example, if querying sensor data, the use of standards – e.g. SensorML[3] – helps the reuse of service ontologies since they can be applied to any service using a similar schema. However any other schema can be integrated with the same results.
- *High level services support*: since services are described as SWS, they inherit all benefits of the underlying SWS execution platform (Section 2.3) and are updated as more features are added to the platform. In other solutions support for composition and discovery is imbedded in syntactic standards themselves, which implies specific parsing features and adding ad hoc reasoning

[1] http://irs-test.open.ac.uk/sgis-dev/
[2] http://www.opengeospatial.org/initiatives/?iid=162
[3] http://vast.nsstc.uah.edu/SensorML/

capabilities to standard software applications, which is time consuming and error prone. Moreover, SWS introduce a minimalist approach in the description of a domain, by modeling the concepts used by Web Services only, and allowing on-the-fly creation of instances when Web Services are invoked (lifting).

- *Support of the Emergency Handling Process*: the conceptual distinction between goal and web services - introduced by WSMO – allows developers to easily design business processes known a priori (e.g. emergency procedure) in terms of composition of goals, and move the (automatic) identification of the most suitable service at run-time.

In the future, a new era of emergency management can be envisaged, in which EMS's 'collaborate' thorough the Internet to provide relevant information in emergency situations through SWS technology. In this way, involved agencies and emergency corps can extend their knowledge about a particular emergency situation making use of different functionalities based on data hold by other agencies which otherwise might not be accessible to them or slow to obtain.

5. Open Challenges

Since SWS technology is young and e-Government is a very complex domain, far from every issue is completely addressed. Among the major remaining challenges identified are:

- *SWS infrastructure.* WSMO is an ongoing research, and some of its main features - e.g. orchestration, non functional properties, and quality of services-based discovery - are still under development. Such aspects are likely to yield very useful results in e-Government.
- *Commercialization* The transition of the currently available systems into a stable and robust infrastructure is one of the major challenges that need to be solved, before a SWS-based solution can be deployed into a productive environment. However, the prototyping development of carefully targeted applications with clear objectives can lead to real-world operational systems.
- *Organizational and social aspects*. The employees of governmental agencies usually perform tasks using well established procedures; the inappropriately-handled introduction of new processes or applications may lead to a reluctance to use them using them. Active participation of stakeholders and end-users in the design and development processes allows developers to deploy applications that respect current procedures and, at the same time, ease the work of staff, leading to improved acceptance.
- *Privacy, Security, and Trust.* These are fundamental requirements in e-Government. At the syntax level, efficient solutions for addressing privacy and security issues already exist or else there is relevant ongoing research. The semantic level should extend the syntactic solutions by ontologically describing security and privacy policies of accessing data and processes. Moreover, trust-based discovery of SWS is a crucial issue, in order to avoid invocation of malicious or unreliable services, for which there are no defined standards by which SWS may expose their policies and trust features. The key to enabling a trust-based selection for SWS lies in a common ontological representation, where Web Service and client perform their trust guaranties and requirements [4].
- *Ease of use of SWS technology in e-Government.* Full integration between e-Government and SWS is not an easy task. In order to address these issues, a more complex semantic layer – i.e. an explicitly e-Government framework - needs to be modelled [6].
- *Standardization.* Currently, there are not reference standards for (semantic) service oriented applications in e-Government. The e-Government community is still debating which approach to follow between, as broadly described options, standardization versus. integration (i.e. focusing on interoperation among several existing approaches). We believe that our approach is open to both solutions and our results may contribute to the investigation of possible standards.

6. Conclusion

The adoption of SWS in e-Government appears to be a natural development. Specifically, SWS technology promises to:

- *Provide added value joined up services:* allowing software agents to create interoperating services transparently to the users, and hence automate integration, reasoning and mediation among heterogeneous data sources and processes available at distinct governmental levels.
- *Enable formalization of government business processes in an unambiguous structure:* allowing the creation of a common understanding of processes, and visualization of the knowledge involved. This could eventually lead to a reengineering of the governmental systems and. simplification of processes.
- *Reduce risk and cost* by: moving from "hard coding" services to reusable functionality, for example through utility computing of shared services (e.g. payment platforms, legal resources, etc.); keeping government organizations' autonomy in the description/management of their domain; increasing flexibility; enabling discovery of new or previously unknown services; aggregating services on the basis of user preferences; providing better service to third-parties and customers; and easily addressing the evolution and change of existing services and scenario.
- *Provide better support to front line* by allowing one-stop, customer focused, and multiple viewpoint access to services and shared information.

However, demonstrating this to the e-Government community will require the achievement of several prerequisites: creation of compelling demonstrators and prototypes, establishing visible standards; stable and mature technology and products; and convincing business cases. Perhaps more importantly, this may provide a way, to address existing barriers and perceptions such as:

- *Trust in automated data sharing*. Governmental organizations are concerned about: (i) ownership, control and quality among service providers; (ii) security, data protection, confidentiality, and privacy issues.
- *Patchy awareness of WS*. Stakeholders are often unclear about the distinction between WS and general services available via Web.
- *Up-front Infrastructure costs* (e.g. investment in WS). Governmental organizations are reluctant to be the pioneers which take the initial financial 'hit', in implementing SWS, as with almost any new technology.
- *Market development* in terms of raising the awareness of potential SWS benefits in e-Government, increasing pilot applications, and promoting the availability of working SWS platforms.

In our work, we successfully established a close collaboration with a large local authority in UK, in order to define a reusable SWS-based framework for deploying real-world applications in the e-Government domain. The aim is to dealing with complex scenarios, by easily interconnecting heterogeneous domains, and allowing governmental agencies to cooperate and consume shared data in an easy way and without a centralized control. Following our approach, we deployed an Emergency Management System. In particular, we stressed all of the aspects associated with the development of SWS-based applications: knowledge acquisition, discovery, composition, and mediation.

We believe that our work may contribute to raising awareness of the potential benefits of SWS in e-Government. Perhaps more importantly, our results may also be used to (i) guide the efforts of new e-Government applications/projects; (ii) influence the e-Government standards environment and the e-Government strategic environment so as to encourage take up of SWS technologies.

7. References

[1] BERNERS-LEE, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(4):34-43, (2001).

[2] DOMINGUE, J., Cabral, L., Hakimpour, F., Sell, D., Motta, E.: IRS-III: A platform and infrastructure for creating wsmo-based semantic web services. In Proceedings of the Workshop on WSMO implementation, Frankfurt, Germany, (2004).

[3] EISENSTADT, M., Komzak, J., Dzbor, M.: Instant messaging + maps = powerful collaboration tools for distance learning. In Proceedings of TelEduc03, Havana, Cuba, (2003).

[4] GALIZIA, S.: WSTO: A Classification-Based Ontology for Managing Trust in Semantic Web Services. In proceedings of 3th International Semantic Web Conference, Budva, Montenegro, (2006).

[5] GOOGLE: Google Web APIs, (2005). http://www.google.com/apis/index.html

[6] GUGLIOTTA, A., Cabral, L., Domingue, J.: Knowledge Modeling for Integrating e-Government Applications and Semantic Web Services. AAAI Spring Symposium "Semantic Web meets eGovernment", AAAI Press SS-06-06, Stanford, California, (2006).

[7] MARK, D.: Cognitive Image-Schemata for Geographic Information: Relations to User Views and GIS Interfaces. In Proceedings of GIS/LIS '89 (pp. 551–560). Orlando, Florida, (1989).

[8] MEDJAHED, B., Bouguettaya, A.: Customized Delivery of E-Government Web Services. Web Services, IEEE Intelligent Systems, 20(6). (2005).

[9] MOTTA, E.: An Overview of the OCML Modelling Language, In Proceedings of the 8th Workshop on Knowledge Engineering Methods and Languages. (1998).

[10] ONTOGOV: Ontogov project. (2004). http://www.ontogov.com

[11] TERREGOV: Terregov project. (2004). http://www.terregov.eupm.net/my spip/index.php

[12] WSMO Working Group, D2v1.0: Web Service Modeling Ontology (WSMO). WSMO Working Draft, (2004). http://www.wsmo.org/2004/d2/v1.0/