A Semantic Web Services GIS based Emergency Management Application

Vlad Tanasescu[1], Alessio Gugliotta[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

1. Introduction

In an emergency situation, relevant information about involved elements is required. This information ranges from demographic data, weather forecasts and sensor data, available transportation means, presence of helpful agents, land use and cover statistics or values, etc. Moreover, the emergency management process is dynamic as it involves several definite steps, described in standard procedures from which the Emergency Officer (EO) should not depart without good reason. Multiple agencies own the relevant data and possess parts of emergency related knowledge.

Exchanging this information by interacting on a personal/phone/fax basis is slow and may even be error prone. Using traditional Geographical Information Systems (GIS) to handle specifically Spatial-Related Data (SRD) is not always satisfactory, since data sources are not always suitable exposed and often present various semantics. In an emergency situation, such barriers are unacceptable and the whish of a more complete interoperability through the network is often expressed[1].

The proposed Emergency Management Application (EMA) is a decision support system based on Semantic Web Services (SWS) technology, which assists the EO in the tasks of retrieving, processing, displaying, and interacting with only emergency relevant information, more quickly and accurately. As a result, the involved agencies become able to extend their knowledge about the emergency situation by making use of different functionalities based on date held by other agencies which otherwise might not be accessible to them or slow to obtain manually.

Our work represents a practical e-Government application, where the stakeholders are the governmental agencies, and the end-users are governmental employees. The application has been designed for the Emergency Planning Department of the Essex

---

[1] e.g. http://www.technewsworld.com/story/33927.html

County Council (ECC) – a large local authority in UK, but can be adopted by other public authorities and rescue corps dealing with emergency response situations.

2. Design Choices and Development Methodology

Any information system can gain advantage from the use of semantics [4]. In GIS, the use of semantic layers, although not yet firmly established, is being investigated in a number of research studies [1], [2], [3]. Having ontologies describing a SRD repository and its functionalities is believed to make cooperation with other systems easier and to better match user needs. In particular, SWS technology may provide an infrastructure in which new services can be added, discovered and composed continually, by combining the flexibility, reusability, and universal access that typically characterize a Web Service, with the expressivity of semantic markup and reasoning. This will allow the automatic invocation, composition, mediation, and execution of complex services with multiple paths of execution, and levels of process nesting. In order to provide semantic and step toward the creation of added value services, we adopt WSMO[2] – a promising SWS framework – and IRS-III [5] – a tested implementation of this standard. The reference language for creating ontologies is OCML [6].

Our development process firstly enables the data and functionalities provided by existing legacy systems to be exposed as Web Services (WS). Then, the latter are semantically annotated and published using IRS-III SWS infrastructure. The following layered architecture of the application reflects and explains this double stage process:

- *Legacy System layer:* consists of existing data sources and IT systems provided by each of the involved governmental parties.
- *Service Abstraction layer:* exposes the functionalities of the legacy systems as WS, abstracting from the hardware and software platforms of the legacy systems. Whenever a new service is available at this layer, it will be semantically described and properly linked to existing semantic descriptions.
- *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 WSMO descriptions, which are composed of goals, mediators, and Web services, supported by relevant domain ontologies. This layer provides the flexibility and scalability of our application. Managing the semantic description, the semantic developer can introduce new functionalities of the application (e.g. new EO goals that can be invoked by the user interface) or updating existing ones.
- *Presentation layer:* is a Web application accessible through a standard Web browser. The goals defined within the previous layer are reflected in the structure of the interface and can be invoked either through the IRS-III API or as an HTTP

---

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

request. The goal requests are filled with data provided by the user and sent to the Semantic Web Service layer.

In our approach, we aimed to obtain a development process that might be *pragmatic* - in order to quickly lead to a working outcome – as well as *flexible* - in order to easily respond to eventually changes/improvements and meet the multiple actors' viewpoints. For these reasons, we followed a prototyping approach composed of the following three straightforward phases: *Requirements capture*, *SWS description*, and *Evaluation*. The last phase triggers the prototyping iterations of the SWS description phase on the basis of involved actors' feedback. At this stage, the application has been shown to the Planning Department Officers and other people dealing with emergency situations in the ECC area (e.g. officers of the London Stansted Airport). Future improvements and changes have been mainly planned on the basis of their feedback, such as accessing to traffic cameras in the affected area.

3. The Emergency Management Application

Following several interviews with SRD holders in ECC, it was decided to focus the application on a real past emergency situation: a snowstorm which affected the M11 motorway on 31st January 2003[3]. To present the application, we follow the layered architecture introduced in the previous section.

Legacy System Layer. The EMA aggregates data and functionalities from three structurally independent and heterogeneous, real world 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. As can be expected ViewEssex contains a wide range of data including data for roads, administrative boundaries, buildings, and Ordnance survey 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* is an Instant Messaging client facilitating lightweight communication, collaboration, and presence management [7] built on top of the instant messaging protocol Jabber[4]. The BuddySpace client can be accessed on standard PCs, as well as on PDAs and on mobile phones which in an emergency situation may be the only hardware device available.

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

- *Meteorological service:* this service provides weather information (e.g. snowfall) over a specific rectangular spatial area.

---

[3] BBC news web site: http://news.bbc.co.uk/2/hi/talking_point/2711291.stm
[4] Jabber. http://www.jabber.org/

- *ECC Emergency Planning services*: using the ViewEssex data each service in this set returns detailed information on a specific type of rest centre within a given circular area. For example, the *'getHospitals'* Web service returns a list of relevant hospitals.
- *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 and make use of the EMA ontologies. In particular, we created a number of filter services that manipulate GIS data according to emergency-specific requirements semantically described (e.g. rest centres with heating system, hotels with at least 40 beds, easier accessible hospital, etc.). The criteria used were gained from our discussions with the EOs.

Domain Ontologies for the Semantic Web Service Layer. The following ontologies reflecting the client and provider domains were developed to support WSMO descriptions:
- *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. If a new source and the Web services exposing its data and functionalities are integrated, a new domain ontology has to be introduced[5]. The services, composed of the data types involved as well as its interface, have to be described in such a ontology usually at a level low enough to remain close from the data.

To get the information provided by web services up to the semantic level, we introduce *lifting operations* that allows the passage of data types instances from a syntactic level (xml) to an ontological one (OCML) specified in the domain ontology definitions. We found that this process can be automated every time the domain ontology one can be.
- *HCI Ontology*: part of the user layer, this ontology is composed of HCI and user-oriented concepts. It allows to lower from the semantic level results for the particular interface which is used (e.g. stating that Google Maps API is used, defining "pretty names" for ontology elements, etc.). Note that although the choice of the resulting syntactic format depends of the chosen lowering process, concepts from the HCI ontology are used in order to achieve this transformation in a suitable way.
- *Archetypes Ontology*: part of the user layer, this is a minimal ontological commitment ontology aiming to provide a cognitively meaningful insight into the nature of a specialized object; for example, 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. It is moreover assumed that

---

[5] Here existing ontologies can be reused.

any client, whilst maybe lacking the specific representation for a specific basic level concept, knows its archetypal representation.

- *Spatial Ontology*: a part of the mediation layer, it describes GIS concepts of location, such as coordinates, points, polygonal areas, and fields. It also allows describing spatial objects as entities with a set of attributes, and a location.

The purpose of the HCI, Archetypes and Spatial ontologies is the aggregation of different data sources on, respectively, a representation, a cognitive and a spatial level. Therefore we can group them under the appellation *aggregation* ontologies. They allow the different data sources to be handled and presented in a similar way. Inversely to the lifting operations, *lowering operations* transform instances of aggregation ontologies into syntactic documents to be used by the server and client applications. This step is usually fully automated since aggregation ontologies are, by definition, quite stable and unique.

- *Context Ontology*: the context ontology allows describing *context n-uples* which represent a particular situation. In the emergency planning application, context n-uples have up to four components, the use case, the user role, the location, and the type of object. Contexts are linked with goals, i.e. if this type of user accesses this type of object around this particular location, these particular goals will be presented. Contexts also help to inform goals, e.g. if a goal provides information about petrol stations in an area, the location part of the context is used to define this area, and input from the user is therefore not needed. Each time an object is displayed by a user at a particular location, a function of the context ontology provides the goals which need to be displayed and what inputs are implicit.

WSMO descriptions for the Semantic Web Service Layer. As introduced in the previous section, the goals, mediators, and Web services descriptions of our application link the Met 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 HCI and archetypes ontologies. Mismatches are resolved by the defined mediators.

A small portion of emergency management process (workflow) represented in terms of SWS descriptions is shown in Figure 1. *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). 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 1 address these problems.

- *Web service discovery:* each SWS description of a 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.
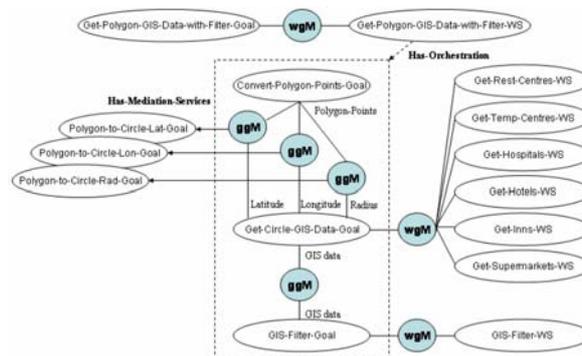
Fig. 1. A portion of the WSMO descriptions for the EMS application.

It is important to note that if new WS – for instance providing further SRD from other 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.

Presentation Layer. The application user interface is based on Web standards. XHTML and CSS are used for presentation and JavaScript is used to handle user interaction together with AJAX techniques to communicate with IRS-III. One of the main components of the interface is a map, which uses the Google Maps API to display polygons and objects (custom images) at specific coordinates and zoom level.

Goals and attributes are attached to such objects; they are displayed in a pop up window or in a hovering transparent region above the main interface.

Although easy to extend, the actual prototype handles only snow storms and hazards emergency types in the context of the Essex County, according to our real past reference scenario. 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 EO describes and emergency situation, before trying to contact relevant agents. The procedure is as follows:

1. The EO clicks within the displayed hazard region to bring up a menu of available goals. In this case (Figure 2a) three goals are available: show available shelters, login to BuddySpace and get the presence information for related staff.
2. The EO asks for the available Rest Centres inside the region, and then inspects the detailed attributes for the Rest Centre returned (Figure 2b).
3. The EO requests to see the presence status for all staff within the region and then initiates an online discussion the closest online agency worker (Figure 2c).
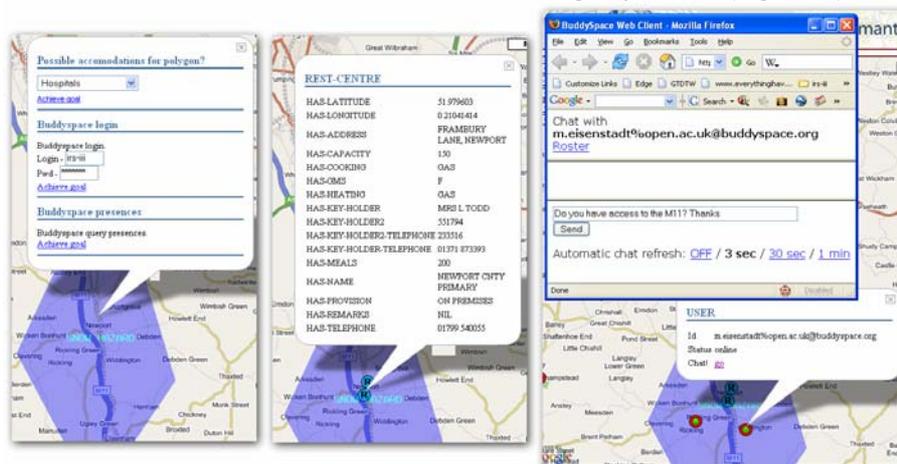
Fig. 2. Showing three screenshots of our application in use.

To a comprehensive description of the operations provided to the user, please refer to our on-line screencast tutorial and live demo[6] .

4. Lesson Learned

In our approach, the integration of new data sources results relatively simple; the steps involved in the process of adding new data sources can be summarized as

---

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

follow: ontological description of the service; lifting operations definition; mapping to aggregation ontologies; goal description; mediation description; lowering definition; and context linking. Although this procedure may seem tedious, and can actually only be performed by a knowledge expert, it presents many advantages compared to standard based approaches as the one demonstrated in the OWS-3 Initiative[7]:

- *Framework openness*: standards are helpful but not necessary. For example, if querying sensor data, the use of standards – e.g. SensorML[8] – helps the reuse of service ontologies and lifting procedures 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, such as discovery and composition, and are updated as more features are added to the platform (e.g. trust based invocation). In other solutions support for composition and discovery is imbedded in syntactic standards themselves, which implies specific parsing features and adding ad hoc reasoning 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*: also, the constant use of context to link goals and situations greatly enhances the decision process. Indeed, actions are oriented depending on the use case, the object, user role and location. With the help of explanations of the utility of each goal in each context, the Emergency Officer's task is greatly simplified. A future development of the context ontology will include feedback from goal invocation history, and allow workflow definitions, i.e. this goal only appears after these two have been invoked. Note that all goals are also accessible independently of any context which allows non directed queries to occur, if needed.

References

1. Casati, R., Smith, B., Varzi, A. C.: Ontological tools for geographic representation. (1998) 77–85.
2. Peuquet, D., Smith, B., Brogaard B.: The ontology of fields. (1999).
3. Fonseca, F. T., Egenhofer, M. J.: Ontology-Driven Geographic Information Systems. ACM-GIS (1999) 14-19.
4. Semantic Interoperability Community of Practice (SICoP). Introducing Semantic Technologies and the Vision of the Semantic Web. (2005).
5. Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: IRS-III: A Broker for Semantic Web Services based Applications. In proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, USA (2006).
6. Motta, E.: An Overview of the OCML Modelling Language. (1998).
7. Eisenstadt, M., Komzak, J., Dzbor, M.: Instant messaging+maps = powerful collaboration tools for distance learning. In proceedings of TelEduc03, Havana, Cuba (2003).

---

[7] http://www.opengeospatial.org/initiatives/?iid=162
[8] http://vast.nsstc.uah.edu/SensorML/