DIP

Data, Information and Process Integration with Semantic Web Services

FP6 - 507483

Deliverable

WP 9: Case Study eGovernment
D9.12
SWS Enhanced GIS prototype (IRSIII) v2.0

Alessio Gugliotta
Vlad Tanasescu
John Domingue
Leticia Gutierrez
Rob Davies
Mary Rowlatt
Jon Bryant
Sandra Stincic
Marc Richardson

September 26th, 2006

EXECUTIVE SUMMARY

This deliverable describes the second version of the SWS enhanced GIS prototype, extending the work introduced in deliverable D9.11: SWS Enhanced GIS prototype (IRSIII) v1.0.

This prototype contributes to the following golden bullet of DIP: Real Use Case Implementation of SWS in the e-Government sector. It represents a SWS-based Emergency Management Application (EMA) and has been developed to fulfil Essex County Council's (ECC) needs to access geospatial data for emergency management and to share it with other partners (e.g.: police, ambulance services, meteorological office, other public organizations, etc) during an emergency situation.

The improvements of the second prototype mainly involve the following two application's architectural layers:

- Ontologies: user-oriented ontologies (now named *aggregation ontologies*) have been improved to better describe concepts associated with the visualization of geospatial objects in the user interface; moreover, a new ontology (named *context ontology*) has been developed to introduce and represent the context of emergency situations within the application.
- User Interface: the old user interface has been integrated with ontology browsing and authoring facilities. This benefits the openness of the application, since expert users can easily get aware of existing semantic descriptions and provide new ones in order to introduce new services (i.e. new data sources and processes). Moreover, the current version of the user interface adopts the DIP APIs for interfacing to IRS-III [1] – the specific WSMO implementation at the basis of this prototype; as a result, the same interface has been reused in the WSMX implementation of the GIS prototype, deliverable D9.13: SWS Enhanced GIS prototype (WSMX) v 1.0.

The deliverable is intended to be read by DIP technical partners to inform them of the way in which the tools/technology has been used to create a use case scenario. In addition, it will be of interest to the end-user community and other data suppliers (e.g.: Essex County Council and other public authorities' emergency planners, police, Essex Fire & Rescue Service, ambulance service, highways and transport, meteorological office, Ordnance Survey, BAA safety services, Essex Rover Rescue).

The deliverable follows the document structure used in deliverable D9.11, but mainly focuses on the improved aspects.

Document Information

IST Project Number	FP6 – 507483	**Acronym**	DIP
Full title	Data, Information, and Process Integration with Semantic Web Services		
Project URL	http://dip.semanticweb.org		
Document URL			
EU Project officer	Brian Macklin		

Deliverable	**Number**	9.12	**Title**	SWS Enhanced GIS prototype (IRSIII) v2.0
Work package	**Number**	9	**Title**	Case study e-Government

Date of delivery	**Contractual**	M 30	**Actual**	M 30
Status	version. 1.0		final	
Nature	Prototype ☑ Report ☐ Dissemination ☐			
Dissemination Level	Public ☐ Consortium ☑			

Authors (Partner)	OU, Essex County Council		
Responsible Author	Alessio Gugliotta	**Email**	a.gugliotta@open.ac.uk
	Partner OU	**Phone**	+44 1908-655790

Abstract (for dissemination)	WP9 is about a case study on e-government. This document provides the WSMO descriptions for the Semantic Web Services developed for the 'GIS prototype'.
Keywords	WSMO descriptions, GIS prototype

Version Log

Issue Date	**Rev No.**	**Author**	**Change**
26.09.2006	001	Alessio Gugliotta	Creates initial version of the Document
	002		
	003		
	004		

Project Consortium Information

Partner	Acronym	Contact
National University of Ireland Galway	NUIG	Prof. Dr. Christoph Bussler Digital Enterprise Research Institute (DERI) National University of Ireland, Galway Galway Ireland Email: chris.bussler@deri.org Tel: +353 91 512460
Fundacion De La Innovacion.Bankinter	Bankinter	Monica Martinez Montes Fundacion de la Innovation. BankInter Paseo Castellana, 29 28046 Madrid, Spain Email: mmtnez@bankinter.es Tel: 916234238
Berlecon Research GmbH	Berlecon	Dr. Thorsten Wichmann Berlecon Research GmbH Oranienburger Str. 32 10117 Berlin, Germany Email: tw@berlecon.de Tel: +49 30 2852960
British Telecommunications Plc.	BT	Dr John Davies BT Exact (Orion Floor 5 pp12) Adastral Park Martlesham Ipswich IP5 3RE, United Kingdom Email: john.nj.davies@bt.com Tel: +44 1473 609583
Swiss Federal Institute of Technology, Lausanne	EPFL	Prof. Karl Aberer Distributed Information Systems Laboratory École Polytechnique Féderale de Lausanne Bât. PSE-A 1015 Lausanne, Switzerland Email : Karl.Aberer@epfl.ch Tel: +41 21 693 4679
Essex County Council	Essex	Mary Rowlatt, Essex County Council PO Box 11, County Hall, Duke Street Chelmsford, Essex, CM1 1LX United Kingdom. Email: maryr@essexcc.gov.uk

		Tel: +44 (0)1245 436524
Forschungszentrum Informatik	FZI	Andreas Abecker Forschungszentrum Informatik Haid-und-Neu Strasse 10-14 76131 Karlsruhe Germany Email: abecker@fzi.de Tel: +49 721 9654 0
Institut für Informatik, Leopold-Franzens Universität Innsbruck	UIBK	Prof. Dieter Fensel Institute of computer science University of Innsbruck Technikerstr. 25 A-6020 Innsbruck, Austria Email: dieter.fensel@deri.org Tel: +43 512 5076485
ILOG SA	ILOG	Christian de Sainte Marie 9 Rue de Verdun, 94253 Gentilly, France Email: csma@ilog.fr Tel: +33 1 49082981
inubit AG	Inubit	Torsten Schmale inubit AG Lützowstraße 105-106 D-10785 Berlin Germany Email: ts@inubit.com Tel: +49 30726112 0
Intelligent Software Components, S.A.	iSOCO	Dr. V. Richard Benjamins, Director R&D Intelligent Software Components, S.A. Pedro de Valdivia 10 28006 Madrid, Spain Email: rbenjamins@isoco.com Tel. +34 913 349 797
The Open University	OU	Dr. John Domingue Knowledge Media Institute The Open University, Walton Hall Milton Keynes, MK7 6AA United Kingdom Email: j.b.domingue@open.ac.uk Tel.: +44 1908 655014
SAP AG	SAP	Dr. Elmar Dorner SAP Research, CEC Karlsruhe SAP AG Vincenz-Priessnitz-Str. 1

		76131 Karlsruhe, Germany Email: elmar.dorner@sap.com Tel: +49 721 6902 31
Sirma AI Ltd.	Sirma	Atanas Kiryakov, Ontotext Lab, - Sirma AI EAD Office Express IT Centre, 3rd Floor 135 Tzarigradsko Chausse Sofia 1784, Bulgaria Email: atanas.kiryakov@sirma.bg Tel.: +359 2 9768 303
Unicorn Solution Ltd.	Unicorn	Jeff Eisenberg Unicorn Solutions Ltd, Malcha Technology Park 1 Jerusalem 96951 Israel Email: Jeff.Eisenberg@unicorn.com Tel.: +972 2 6491111
Vrije Universiteit Brussel	VUB	Carlo Wouters Starlab- VUB Vrije Universiteit Brussel Pleinlaan 2, G-10 1050 Brussel ,Belgium Email: carlo.wouters@vub.ac.be Tel.: +32 (0) 2 629 3719

ACRONYMS/GLOSSARY

BAA British Airports Authority

ECC Essex County Council

EMS Emergency Management System

EPO Emergency Planning Officer

GIS Geographical Information Services

IRSIII Internet Reasoning Server 3

OU Open University

SRD Spatially Related Data

SWS Semantic Web Service

WP9 Work Package 9

WgM Web Service-Goal mediator

WSMO Web Service Modelling Ontology

TABLE OF CONTENTS

EXECUTIVE SUMMARY I

ACRONYMS/GLOSSARY VI

TABLE OF CONTENTS VII

1 INTRODUCTION 9

2 ARCHITECTURE 9

- 2.1 Ontologies for the Semantic Web Service layer 10
- 2.2 WSMO descriptions for the Semantic Web Service Layer 11

3 PRESENTATION LAYER 13

4 ON-LINE DEMO 15

5 CONCLUSIONS 15

REFERENCES 17

LIST OF FIGURES

Figure 1 - The general architecture of the EMA. 9
Figure 2 - A portion of WSMO descriptions for the EMA. 12
Figure 3 - The "new look" of the prototype's interface. 14
Figure 4 - Direct link from goals to ontologies. 14
Figure 5 - Ontology browsing for new area type attribution. 14

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.

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.

2 ARCHITECTURE

Figure 1 depicts the multi-layered architecture of the proposed EMA.

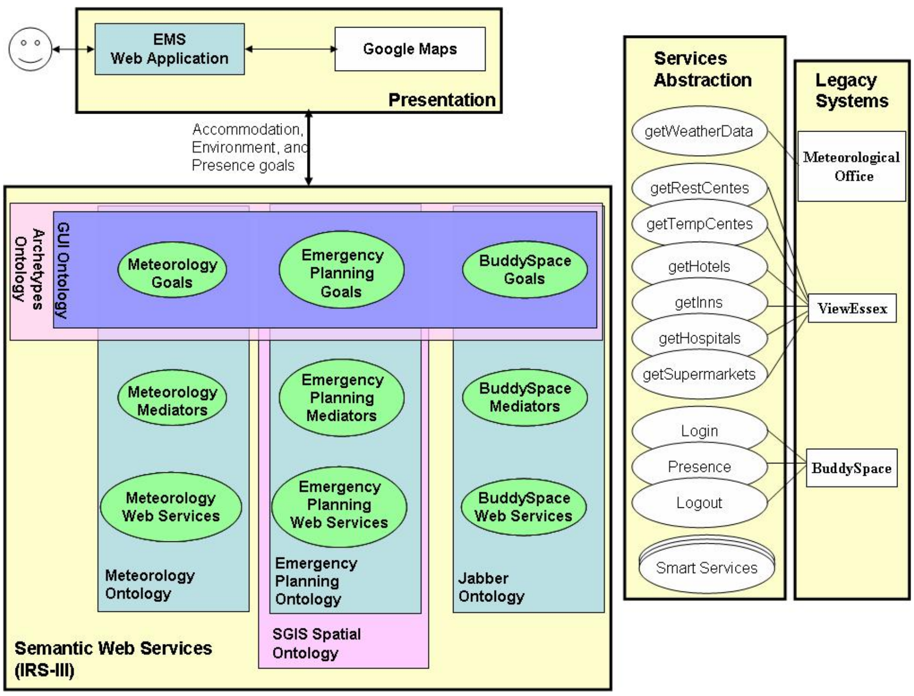

Figure 1 - The general architecture of the EMA.

The application is composed of the following four layers:

- *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 [1], 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 request. The goal requests are filled with data provided by the user and sent to the Semantic Web Service layer.

The second version of the prototype improved the Semantic Web Service and Presentation layers and did not integrate new data sources and services. For this reason, the following sections detail the modified layers only. For a description of the other two layers please refer to D9.10 and D9.11.

2.1 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 - also reusing existing ontologies. 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 (class instances) specified in the domain ontology definitions. These LISP functions automatically extract data from SOAP messages and create the counterpart class instances. The mapping information between data types and ontological classes is defined at design time by developers.

- *GUI Ontology*: part of the user layer, this ontology is composed of GUI and user-oriented concepts. It allows to lowering 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 GUI 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 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 GUI, 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.

- *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.

2.2 WSMO descriptions for the Semantic Web Service Layer

For illustration purposes, a small portion of the SWS descriptions are shown in Figure 2. The example details the main goal “*Locate suitable shelters for evacuated people*”. For a more detailed description of WSMO descriptions please refer to D9.10.

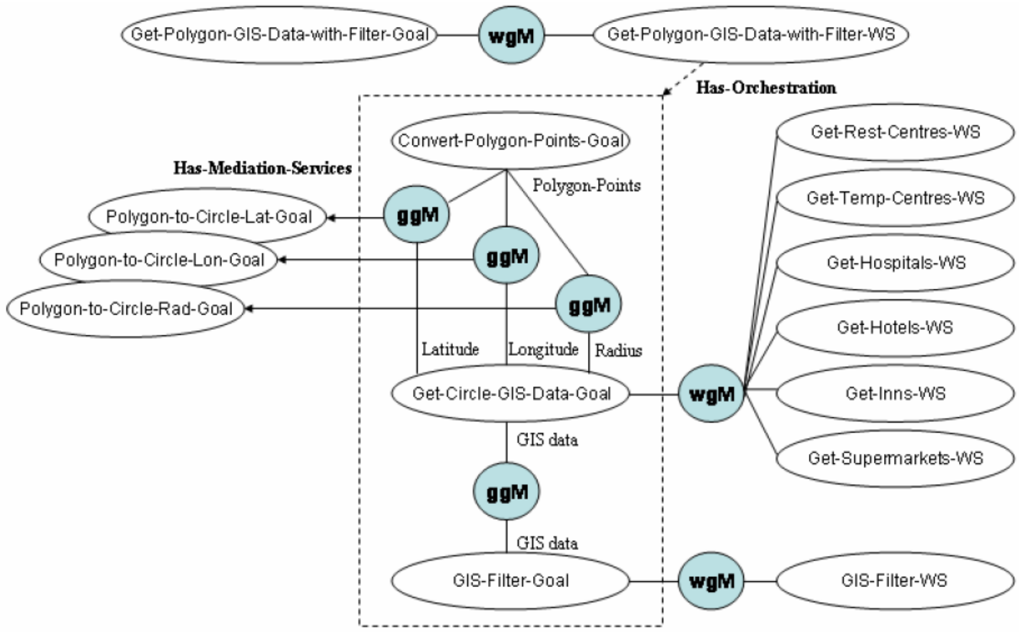

Figure 2 - A portion of WSMO descriptions for the EMA.

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 4 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 (depicted 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 Web Services – 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.

3 PRESENTATION LAYER

The presentation layer has been entirely refractored since the previous version (see **Error! Reference source not found.** Indeed, the use of disparate web technologies (JavaScript to implement AJAX, HTML + JavaScript for the user interface) made the prototype dependent of the browser at runtime (cross browser functionality was difficult to achieve) as well as at development time (JavaScript debugging could only occur on the host browser). Moreover, given the absence of strong typing in JavaScript, the addition of new functionalities was becoming problematic as the size of the presentation layer was increasing. GWT, the Google Web Toolkit[1], allows eliminating these concerns by providing a pure Java development environment; the front end is written in the Java programming language, and the GWT compiler converts Java classes to browser-compliant JavaScript and HTML.

Functionalities in version 2 of the prototype are comparable to version 1, i.e. it allows defining areas which are given a type, from the emergency ontology, types which present goals that can be invoked in order to retrieve contextually relevant emergency related elements. However, the rewriting takes advantage of the functionalities of Google Maps 2, which includes maps for almost all Europe, and with which version 1 of the prototype was incompatible.

Also the integration with IRS and the underlying ontologies is much tighter. As an example, when a goal is selected for an object, the user interface for this goal is directly extracted from the goal definition, and instances are fetched whenever an input role is not of a basic type (**Error! Reference source not found.**). In a similar way, the interface allows the user to browse the ontology in order to attribute a type to a new area, i.e. polygon (**Error! Reference source not found.**

However, concerns have been raised about the reduced user-friendliness induced by the usage of ontology names. This issue is about to be solved by adding to the GUI ontology a mapping between ontology types and their user names. We believe that this technique, coupled with GWT's support for internationalisation, will allow us to obtain a multiple language interface.

[1] http://code.google.com/webtoolkit/

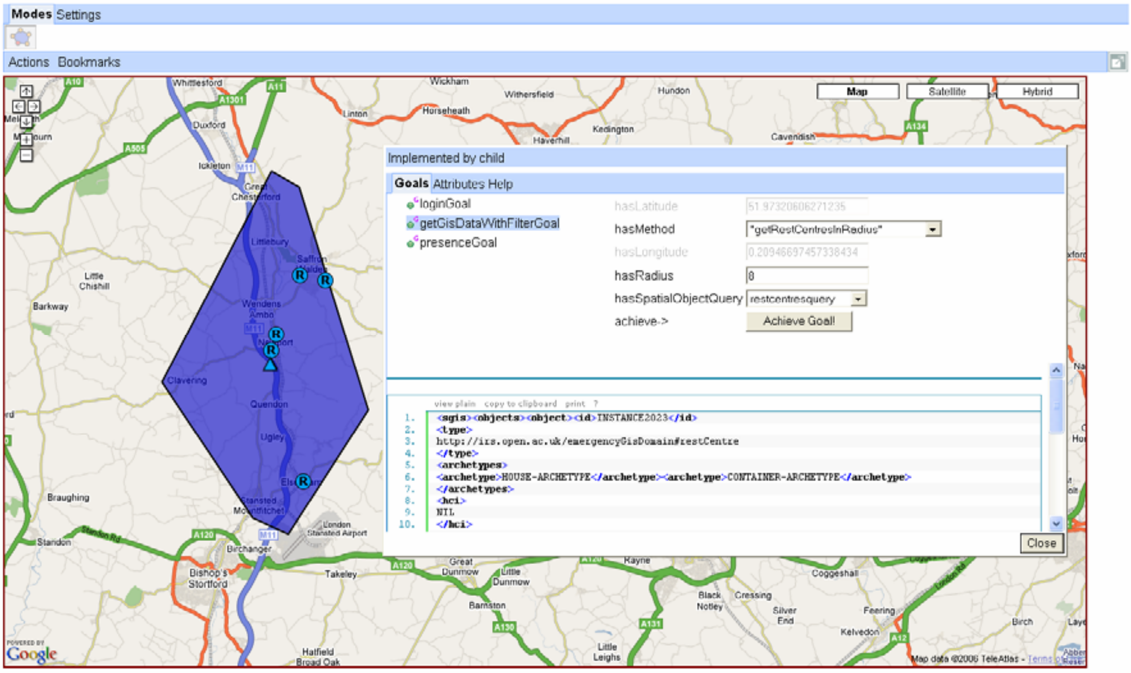

Figure 3 - The "new look" of the prototype's interface.

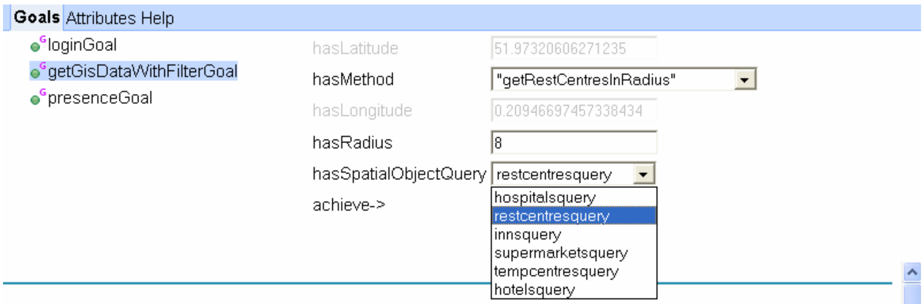

Figure 4 - Direct link from goals to ontologies.

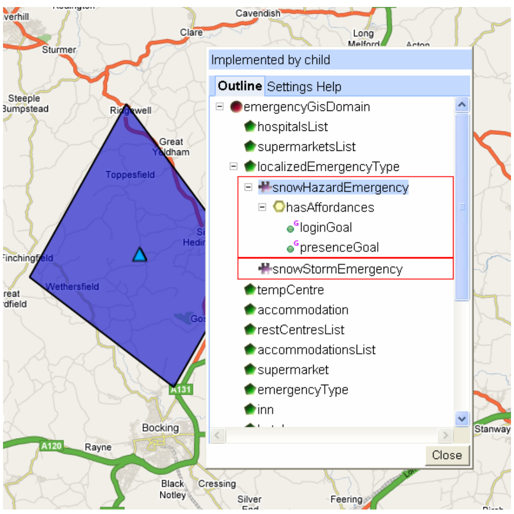

Figure 5 - Ontology browsing for new area type attribution.

4 ON-LINE DEMO

A screencast of the interaction as well as a live version are available online[2].

5 CONCLUSIONS

This described prototype adopts WSMO and IRS-III to (i) provide an infrastructure, in which new services can be added, discovered and composed continually; (ii) allow the automatic invocation, composition, mediation, and execution of complex services.

The integration of new data sources is relatively simple although not totally trivial. The steps involved in the process of adding a new data source, as well as the ability to automate them, are described in the following:

- *Ontological description of service*: the service, composed of the data types involved as well as its interface has to be described in a low level ontology, usually at a level low enough to remain close from the data. This step can be automated in many cases.
- *Lifting definition*: the lifting operation allows the passage of data types instances from a syntactic level (xml) defined in the data schema to an ontological one (ocml) specified in the ontology definition. We found that this process can be automated every time the previous one can be.
- *Mapping to integration ontologies*: this process can be fully automated by default, and customizable as needed.
- *Goal description*: a new goal has to be defined which represents the newly integrated web service.
- *Mediator description*: the goal has to be linked to the WS with a mediator, which is often a trivial operation.
- *Lowering definition*: the lowering operation transforms instances of aggregation ontologies into syntactic documents to be used by the server and client applications.

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[3]:

- *Framework openness*: standards are helpful but not necessary. For example, if querying sensor data, the use of standards – e.g. SensorML[4] – 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 and are updated as more features

[2] http://irstest.open.ac.uk/sgis-dev/.

[3] http://www.opengeospatial.org/initiatives/?iid=162

[4] http://vast.nsstc.uah.edu/SensorML/

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*: 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. Specifically, 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] **IRS-III: A Broker for Semantic Web Services based Applications.** Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: In proceedings of the 5th International Semantic Web Conference (ISWC 2006), Athens, USA (2006).