

Article

A Semantic Sensor Web for Environmental Decision Support Applications

Alasdair J. G. Gray ^{1,*}, Jason Sadler ², Oles Kit ^{2,3}, Kostis Kyzirakos ⁴, Manos Karpathiotakis ⁴, Jean-Paul Calbimonte ⁵, Kevin Page ⁶, Raúl García-Castro ⁵, Alex Frazer ⁶, Ixent Galpin ¹, Alvaro A. A. Fernandes ¹, Norman W. Paton ¹, Oscar Corcho ⁵, Manolis Koubarakis ⁴, David De Roure ⁶, Kirk Martinez ⁶ and Asunción Gómez-Pérez ⁵

¹ School of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, UK; E-Mails: I.Galpin@cs.man.ac.uk (I.G.); A.Fernandes@cs.man.ac.uk (A.F.); N.Paton@cs.man.ac.uk (N.P.)

² GeoData Institute, University of Southampton, Southampton SO17 1BJ, UK; E-Mails: jds@geodata.soton.ac.uk (J.S.); oyk@geodata.soton.ac.uk (O.K.)

³ Department of Climate Impacts and Vulnerabilities, Potsdam Institute for Climate Impact Research, P.O. Box 60 12 03, D-14412 Potsdam, Germany

⁴ Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens 15784, Greece; E-Mails: kkyzir@di.uoa.gr (K.K.); mk@di.uoa.gr (M.K.); koubarak@di.uoa.gr (M.K.)

⁵ Ontology Engineering Group, Universidad Politécnica de Madrid, Campus de Montegancedo s/n 28660, Boadilla del Monte, Madrid, Spain; E-Mails: jpcalbimonte@delicias.dia.fi.upm.es (J.-P.C.); rgarcia@fi.upm.es (R.G.-C.); ocorcho@fi.upm.es (O.C.); asun@fi.upm.es (A.G.-P.)

⁶ School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; E-Mails: krp@ecs.soton.ac.uk (K.P.); ajf3@ecs.soton.ac.uk (A.F.); dder@ecs.soton.ac.uk (D.D.R.); km@ecs.soton.ac.uk (K.M.)

* Author to whom correspondence should be addressed; E-Mail: A.Gray@cs.man.ac.uk; Tel.: +44-161-275-0145; Fax: +44-161-275-6204.

Received: 26 July 2011; in revised form: 29 August 2011 / Accepted: 29 August 2011/

Published: 14 September 2011

Abstract: Sensing devices are increasingly being deployed to monitor the physical world around us. One class of application for which sensor data is pertinent is environmental decision support systems, e.g., flood emergency response. For these applications, the sensor readings need to be put in context by integrating them with other sources of data about

the surrounding environment. Traditional systems for predicting and detecting floods rely on methods that need significant human resources. In this paper we describe a semantic sensor web architecture for integrating multiple heterogeneous datasets, including live and historic sensor data, databases, and map layers. The architecture provides mechanisms for discovering datasets, defining integrated views over them, continuously receiving data in real-time, and visualising on screen and interacting with the data. Our approach makes extensive use of web service standards for querying and accessing data, and semantic technologies to discover and integrate datasets. We demonstrate the use of our semantic sensor web architecture in the context of a flood response planning web application that uses data from sensor networks monitoring the sea-state around the coast of England.

Keywords: semantic sensor web; application and visualisation; semantic data integration

1. Introduction

Sensor networks are increasingly deployed to monitor the state of the physical environment around us. Over time, a fuller picture of the environment can be built up by analysing the historic values sensed with these devices, and relating these to the dynamically changing current values, thus enabling a better understanding of both current and evolving conditions. For example, consider the benefits of being able to forecast the severity of tidal surges, and the resulting flooding, which have the potential for devastating effects to business and lives. To effectively predict when a surge is going to take place requires gathering data from a wide variety of sources published by independent autonomous providers: sensor networks that monitor the status of the sea provided by research institutions, government agencies, and private companies; weather forecasts provided by national meteorological offices, and companies; and coastal defence information provided by government departments. These are used as inputs to environmental models which predict the future sea-state, and the probabilities that sea defences will be breached or over-topped. Moreover, planning the response to a potential flooding event requires a large number of additional sources to be available (e.g., shipping, traffic, and man-made assets), which can be related to the results of the forecast and the current conditions.

An extensive review of advances in geosensor networks [1] identified the need to integrate sensor network data with existing, large-scale sensors such as remote sensing instruments or large, stationary ocean buoys. Processing the information in real-time using a data streaming paradigm was also stated as a major challenge for geosensor networks. The idea of a *sensor web*, which enables the interoperability of sensor data to support re-use of existing sensor networks, and relating the sensor data with stored data (*i.e.*, historic and contextual data in databases) and graphical sources (e.g., maps, raster, vector), aims to meet these challenges. More broadly, the key features of a sensor web architecture are the ability to:

- (1) identify relevant sources of data;
- (2) access sensor data in near real-time together with contextual data;

- (3) combine and correlate data from disparate sources with differing modalities (*i.e.*, a stream of sensor data with contextual data stored in a database); and
- (4) enable users and data providers to work with their conceptualisation of the data, *i.e.*, not force users and data providers to use a common data model, particularly as data sources will not be under the control of the users, and could already be publishing their data according to their own conceptualisation.

Similar issues were identified in a recent vision paper [2]. Specifically, the common goals include: support for reusing data originating from autonomous sources; a service infrastructure that supports multiple usage paradigms, viz. request/response, event driven, and resource-oriented; and to enable context sensitive applications that seamlessly integrate environmental monitoring.

In this paper, we argue for the use of semantic web technology to supply rich metadata in a machine understandable format together with a set of services to exploit these semantic annotations to address these issues. We describe a sensor web architecture that enables integration and correlation of multiple heterogeneous datasets, including both live sensor data, historic sensor data, and contextual data (e.g., details of sea defences as provided by the UK Environment Agency's National Flood and Coastal Defence Database (NFCDD) [3], or the road network as a map overlay). We propose an approach that makes extensive use of semantic technologies (*i.e.*, ontologies [4], RDF [5], SPARQL [6]), both for data discovery and data integration. This enables users to interact with data within a conceptualisation that they are familiar with. On the other hand, it does not tie the system to one particular model over the data. Indeed, publishers are free to publish their data according to the model in which it is generated. The publisher could already be publishing their data in a particular model. We aim to support the reuse of data in ways that transcend its original purpose.

The services of the semantic sensor web approach build on existing web service recommendations for data access and integration [7], and notifications [8]. Our approach is complementary to the Open Geospatial Consortium Sensor Web Enablement (OGC-SWE) framework [9,10] in that existing OGC-SWE services (e.g., Sensor Observation Service) can be incorporated into our approach, as well as being used as services on top of our semantic sensor web (as will be described in Section 4).

We demonstrate the use of our semantic sensor web architecture in the context of a flood response planning web application. The application is aimed at a variety of user groups from members of the public to those responsible for identifying and responding to coastal flooding events. The application uses the semantic registry service to discover and dynamically incorporate relevant sources of sensed, modelled, and contextual data according to the user's role and area of interest. Stored and streaming data services are used to access historic and near real-time sensor data, including sources which may have been semantically combined through data integration services. For example, the application provides an innovative tool for alerting users to the incidence and forecast of flood defence over-topping events.

Section 2 provides details of related work in the areas of sensor web architectures and decision support systems. We motivate the use of ontologies for modelling data in the semantic sensor web in Section 3, and present the network of ontologies used in the flood warning system deployment. In Section 4 we provide a detailed description of the services that form the semantic sensor web and their interactions. Section 5 describes the mechanisms for developing environmental decision support applications that mash-up data from several data sources. Our conclusions are given in Section 6.

2. Related Work

In this section we present related work in three areas. First we present related sensor web architectures. We then describe other approaches to providing a semantic sensor web. Finally, we present the state of the art in decision support applications.

2.1. Sensor Web Architectures

The term *sensor web* was originally used to describe a wireless sensor network architecture in which the nodes of the network were autonomous and able to react to the data measured by themselves and other nodes in the network [11]. Since then, the term *sensor web* has been used more generally, as it is in this paper, to describe a distributed web service architecture for publishing, discovering, and combining data from multiple sensor networks and related data sources.

One proposal for such a sensor web architecture is the set of data model definitions and web service specifications that comprise the Open Geospatial Consortium Sensor Web Enablement (OGC-SWE) framework [9,10]. The three data model standards—Observations and Measurements Schema (O&M) [12], Sensor Model Language (SENSORML) [13], and Transducer Markup Language (TML) [14]—provide syntactic data models for representing sensor measurements, the sensors that capture the measurements, and the processing performed on the measurements, respectively. The web service specifications define a service-oriented architecture that provides the functionality to interoperate with sensors and their data across organisation boundaries over the Internet. The Sensor Observation Service (SOS) [15] provides the means by which sensor data can be published, allowing other services and applications to request sensor data. The Sensor Planning Service (SPS) [16] enables, where it is permitted by the sensor, new tasks to be passed to the sensor. The Sensor Alert Service (SAS) [17] and Web Notification Service (WNS) [18] provide mechanisms by which services, applications, and users can receive alerts regarding sensor readings.

A reference implementation of the OGC-SWE framework was developed in the SANY project [19,20]. The SANY project provided SOAP bindings for the abstract service definitions of the OGC-SWE specifications. This enables clients to interact with services using standard web service messages (viz. SOAP) encoded in XML. The SANY services also provided a REST interface [21] to the services. This enables clients to interact with the services through a HTTP interface offering the operations GET, POST, PUT, and DELETE. (See [22,23] for a comparison between REST and SOAP web services.) However, issues of providing a REST-style interface, which focus on resources rather than operations, to a service-oriented architecture were not addressed in the SANY project, see Section 4.4 for more details. A central notion of the SANY project was the “plug and measure” paradigm whereby sensor networks could simply be deployed in the field and the measurements returned through a standard service interface. However the approach relies on the metadata to describe the sensor deployment having already been stored in the system. The SANY project also defined a Cascading Sensor Observation Service for merging data from different sources and performing some post processing of the data, e.g., converting units or computing moving averages [24].

The OGC-SWE architecture, and SANY implementation, enables interoperability between disparate sensor networks by enforcing a common model and syntax for the data: data is exchanged as XML

according to the XML Schema models defined by the OGC-SWE standards. The semantic sensor web architecture proposed in this paper supports the independence of the data sources by allowing them to publish their own data models: a particularly important consideration when data is already being generated by sensors owned by autonomous institutions, or being operated by experts from different domains (each of which has its own set of terminology and concepts). The SANY Cascading SOS supports the merging of data from SOS sources. However, it relies on the data conforming to the standard syntactic encoding, O&M GML. The semantic integrator in the architecture described in this paper supports the merging and correlation of data from disparate sources, including those external to our framework (e.g., SOS). This approach to reuse transcends the original purpose for collecting the data or our set of services.

The OGC-SWE framework does not define a registry service. One option for incorporating such functionality is to use the OpenGIS catalogue service [25]. The OpenGIS catalogue service is based on an **attribute-value** pair data model over which filters, expressed as **attribute-operator-value** statements, can be applied. In the semantic sensor web architecture proposed, both the data model (viz. RDF) and the query language (e.g., stSPARQL) provide a greater level of expressivity. For example, semantic terms drawn from an ontology can be used to refer to spatial geometries besides using coordinates to explicitly specify them.

The European projects OSIRIS and GENESIS have developed a Sensor Instance Registry (SIR) that offers operations for inserting, harvesting and querying information about sensor instances [26]. SENSORML is used to describe sensor instances and queries are posed using a combination of spatial constraints, temporal constraints and keywords. Query answering is carried out by using three indices, one for each of the three kinds of criteria allowed. There is another component called Sensor Observable Registry (SOR) where semantic information about phenomena observed by sensors can be stored using an ontology. SIR can use SOR to exploit semantic information to offer better answers to queries, e.g., to determine equivalent phenomena. A discussion of how to link this discovery framework with OGC catalogues and automatically expose resources provided through OGC-SWE services to them is provided in [26,27].

OGC-SWE also enables the tasking of services using their own mechanism provided through the Sensor Planning Service. We adopt the approach of declarative queries that describe data needs. These are passed down to the sources which can then satisfy the need according to their abilities to answer queries. For example, a query over a sensor network which is able to perform in-network processing can distribute the query into the network to save energy [28].

The Sensor Alert Service and the Web Notification Service provide a custom set of operations to provide limited support for publish/subscribe notifications of events. The semantic sensor web architecture proposed in this paper builds upon the Web Services Base Notification standard (WS-N) [8]. In addition stream query processing languages, such as SNEEql [29] and SPARQL_{Stream} [30], provide support for push-based delivery of sensor measurements according to a declarative description of the data need. Note that the specification of the event service in OGC-SWE v2.0 [31] takes a similar approach to the one proposed in this paper. However, our approach uses a single service to publish sensor data for either polling or push-based delivery whereas the OGC-SWE 2.0 approach requires two services, the SOS and the eventing service.

2.2. Semantics in the Sensor Web

The term *Semantic Sensor Web* was coined by Sheth *et al.* [32], who proposed to combine the OGC-SWE framework with existing W3C standards for annotating service interfaces and publishing sensor data as RDF. After this pioneering work, many other works have embraced the idea of applying semantic technologies for different tasks related to the discovery and integration of data stemming from autonomous heterogeneous data sources, including the work that we present in this paper. The work presented in [33] uses semantic descriptions of sensor deployments to support discovery and syntactic conversion of the data from sensors for publication into an OGC-SWE sensor web. However, the semantic descriptions are not published in the sensor web, which uses the syntactic approaches of the OGC-SWE framework for data discovery and publication.

In [34], five challenges are identified for the application of semantics to the Sensor Web. The first of these challenges is the abstraction level in which sensor data can be obtained, processed and managed in general. The second is the need to handle adequately the quality of data. The third challenge is the integration and fusion of data coming from heterogeneous and autonomously deployed sensor networks. The fourth challenge is the identification and location of relevant sensor-based data sources. The final challenge is the rapid development of applications that are based on these types of data sources. These five challenges are addressed by the semantic sensor web presented in this work.

There is also some ongoing work addressing the publication of sensor data on the Web, in the form of linked sensor data [35,36], although no standardisation or set of agreed best practices have been achieved so far in this respect.

2.3. Decision Support Systems

Computer based *Decision Support Systems* (DSSs) are information systems to enable users to make informed choices based on data and forecasts from a wide variety of sources [37]. DSSs support a wide range of functionality and capabilities to help planners and managers to assess alternatives and ultimately take appropriate decisions. Increasingly, DSSs are offered as web-based or mobile applications. These offer a considerable advantage over traditional DSSs in terms of better cooperation capabilities, access to distributed data sources [38], or increased usability [39]. A number of web-based DSSs applications have been developed to use data stemming from the OGC-SWE framework [31]. However, these are limited by the capabilities of the OGC-SWE framework, as described above, and cannot exploit semantic information provided with the data for discovery, integration or linking to related datasets.

The DHI Group have developed Flood Watch [40] and the Mike modelling framework [41] to provide a DSS in the context of coastal flooding. These proprietary tools provide real-time forecasts in areas prone to flooding and issue early warnings to flood response managers and the public. They can be used to manage and examine real-time data from a range of external sources, support a range of modelling tools, and include a scenario management tool aimed at carrying out comparative assessments and response planning. Being a closed-source system, however, this framework requires extensive customisation and user input at all levels of the decision making process. The ability to dynamically discover relevant data sources, provided by the semantic sensor web architecture proposed in this paper, and the reliance of a light web-based client architecture instead of a centralised approach is certainly beneficial from the

point of view of stakeholder cooperation, e.g., in the emergency response scenario used to motivate our approach. The use of ontologies as a means of mutual understanding between different types of system users in the development and deployment of decision support systems, advocated by this paper, has been successfully tested by the developers of semantically enabled SoKNOS emergency management system [42].

Casola *et al.* [43] describe the architecture of a spatially and sensor type-constrained Early Warning System which combines data from multiple sensor networks and low-level services. The architecture consists of a data modelling and access service, a computing and simulating service, and an alert and notification service. The proposed semantic sensor web architecture aims at ensuring interoperability of different sensor platforms together with contextual datasets and services that operate on the published data, and therefore has the potential to provide the decision makers with more flexible solutions than other highly-customized early warning systems. Jirka *et al.* [44] identify the importance of accessing multi-domain semantically-enabled sensor information for efficient risk monitoring in a number of use scenarios. Our semantic sensor web provides the richer functionality for discovery, data access, and integration identified by Bröring *et al.* [31] as a requirement for disaster management or early warning systems.

3. Modelling Semantic Sensor Web Information

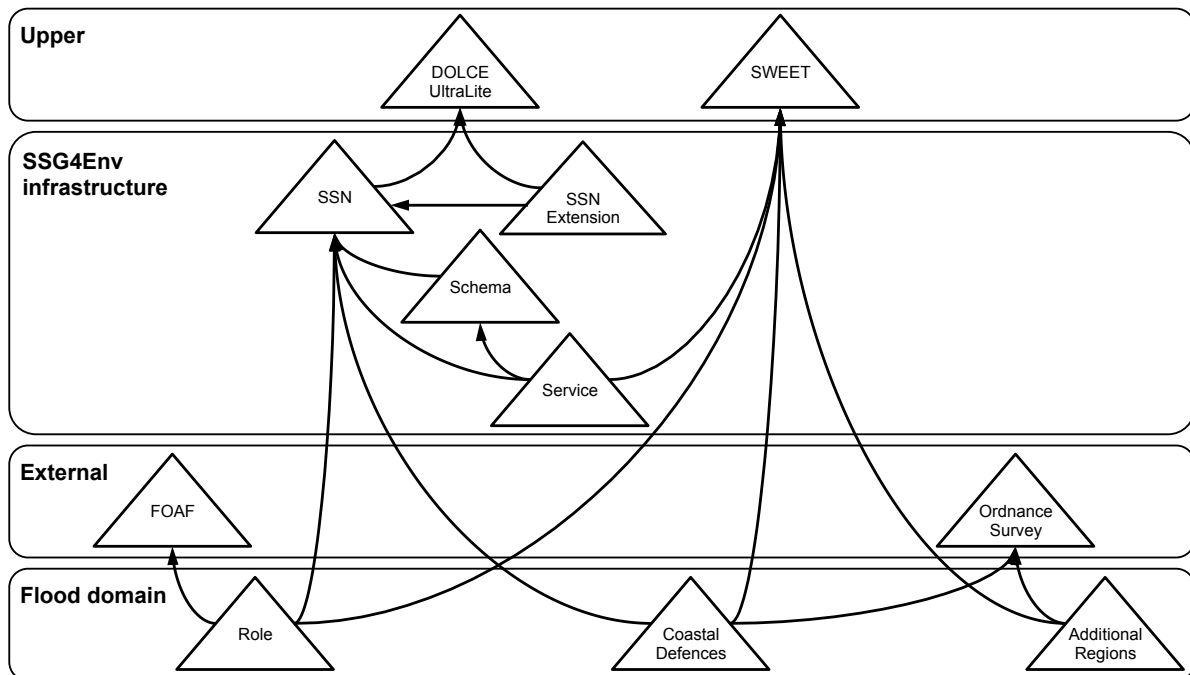
A sensor web provides an information space to enable users to share and manipulate sensor and related data. This assumes that users, and publishers of sensor and related data, can express the meaning of the data according to some model of the real world. The OGC-SWE approach requires all data publishers and users of the sensor web to use the O&M data model that represents data in XML according to a published XML schema. However, this approach requires that existing data sources, e.g., the channel coastal observatory (CCO) sensor network [45] or the national flood and coastal defences database (NFCDD) [3], to map and transform their data to this model rather than publishing their own existing conceptualisation. It does not allow for all the forms of heterogeneity that exist in the sensor web setting, or for the representation of the relationships of the data models of related data sources. It also does not facilitate automated reasoning to classify the data.

A requirement for a sensor web is to support the *ad hoc* responsive evolving use of an information space. The data resources of the information space will contain various forms of heterogeneity, including data modality (*i.e.*, sensed, stored, and graphical), data model (*i.e.*, terminology), and data representation (e.g., relational, RDF, and XML). These need to be reconciled into a coherent conceptualisation for a specific user need. (Note that different conceptualisations are likely to be required for different applications.) We use ontologies to represent the common data model for the information space since they facilitate: (i) describing the different infrastructure services and data sources as well as any domain-dependent information; (ii) having a shared vocabulary to interoperate both across the internal infrastructure services, and between that infrastructure and external sources that adopt alternative approaches, e.g., OGC-SWE based ones [9]; and (iii) discovering, accessing, and integrating information that is shared within the infrastructure.

Figure 1 illustrates the ontology network that is used in the flood emergency planning scenario [46]. The ontology network is composed of different ontologies classified in different layers according to

whether the ontology can be used to represent: domain-specific information required for the scenario, information required for the infrastructure, or upper-level information used to facilitate interoperability among the other ontologies. These ontologies satisfy the following knowledge representation requirements extracted during the development of the architecture and of the scenario prototype.

Figure 1. Ontology network for the flood emergency planning scenario. The arrows indicate ontology reuse. Only the bottom layer of ontologies is specific to the flood application domain.



- To represent sensor networks and their observed information about properties of certain features of interest. This is covered by the *SSN* ontology [47], developed by the W3C Semantic Sensor Network Incubator Group [48]. The *SSN* reuses the *DOLCE+DnS UltraLite* upper ontology [49]. The *SSN* is broadly based on the OGC-SWE O&M model.
- To represent observation collections (included in the OGC-SWE O&M specification but not in the *SSN* ontology), summary data for observation collections, measurement property values, and units of measurement. This is covered by the *SSNExtension* module, an extension of the *SSN* ontology that covers these requirements.
- To represent schema metadata about relations and relational streams. This is covered by the *Schema* module that extends, and corrects, an ontology for relational data and schema components [50].
- To represent the web services provided by the infrastructure and the datasets they provide access to. This is covered by the *Service* module that reuses the *SWEET* upper ontologies [51] and includes concepts from the ISO19119 standard on geographic information services [52]. Note that the *Service* ontology could be related to semantic web service ontologies such as OWL-S [53] or WSMO [54] to support the automated orchestration of services.

- To represent the geographic and administrative regions of the south coast of England. This is covered by the *Ordnance Survey* ontologies [55], which include the regions from Great Britain, and by the *Additional Regions* ontology, which includes other regions needed in our scenario.
- To represent those features of interest and their properties that are specific to the flood emergency planning scenario. This is covered by the *Coastal Defences* ontology.
- To represent the different roles involved in a flood emergency planning scenario. This is covered by the *Roles* ontology.

All the ontologies have been implemented in the OWL ontology language [4], using an ontology engineering tool, and are published on the Web [46]. These ontologies may be updated to meet new requirements in the future; either by upgrading the model represented in the ontology (in the case of ontologies developed by us), or by replacing an ontology with a newer version (in the case of external ontologies). In both cases, new requirements will lead to a new ontology development cycle that will end with new versions of the affected ontologies.

Regarding the adaptability of the ontologies presented here, while some of them are specific to the flood warning scenario, e.g., *Role*, the architecture proposed in the next section is generic. Thus, it can be adapted to other situations by replacing the flood domain ontologies. Note that if an ontology exists for the new application domain, then it is a straightforward process to replace the flood domain ontologies with the ontology for the new domain. Otherwise, a new ontology would need to be developed. This requires a domain expert, who knows the terminology and relationships of the concepts in the domain, to work with an ontology engineer.

4. Sensor Web Architecture

This section defines a semantically enabled sensor web architecture (depicted in Figure 2). The architecture supports:

- (1) discovery of data sources and services based on their content and spatiotemporal coverage;
- (2) accessing and manipulating sensor and related data in near real-time;
- (3) on-the-fly integration of multiple heterogeneous sensor and stored data sources; and
- (4) multiple conceptualisations of data.

The core service types of the architecture (viz. Data Source, Semantic Registry, and Semantic Integrator) form a service-oriented architecture which supports orchestrations that combine the functionality of services, *i.e.*, service instances are used as building blocks to construct more feature-rich interactions of the services. These services expose their capabilities in terms of a collection of reusable interfaces (specified in Table 1), which have been defined using WSDL [56] and interchange SOAP messages [57].

The Application Tier of the architecture provides more domain-specific (e.g., running forecast models over the data as will be described in Section 5.1) or user-oriented functionality (e.g., transforming the presentation of data into a form that is easily processed by user applications). The latter case aims at supporting the increasing trend in decision support systems for rapid development of web-based applications, including mash-ups. Typically, web application developers expect services to offer a

REST interface [21], and potentially a SPARQL endpoint [6]. The service definitions do not preclude implementations from exposing such an interface. However, the High-Level Application Programming Interface (HLAPI) service is defined so that it can be configured to exploit the ability to orchestrate the core services whilst exposing a REST interface.

Figure 2. The service types, and their interactions, that form a semantic sensor web. Note that the Data Source service type collectively represents the three types of data service (see Section 4.2).

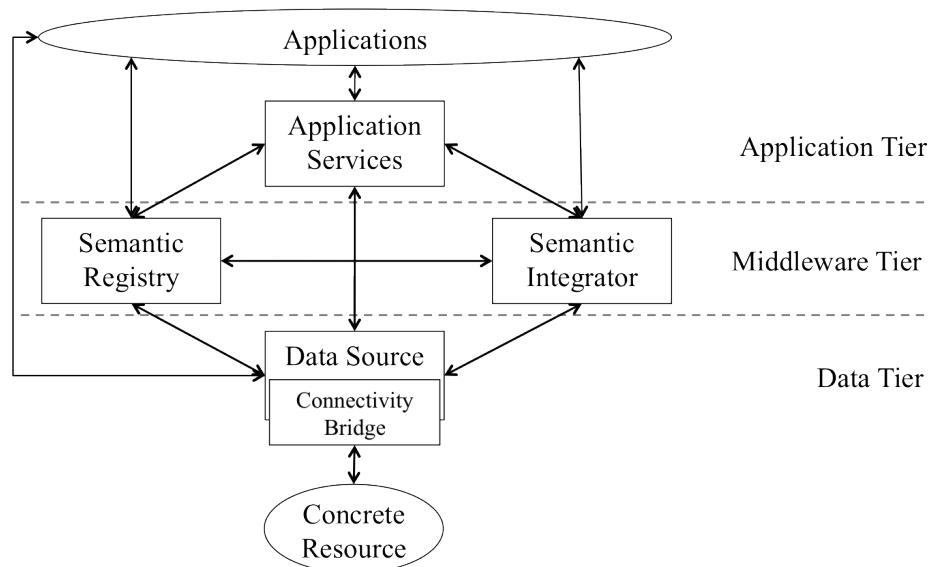


Table 1. Functionality provided by the service interfaces. The use of existing standard interfaces is depicted in brackets. (Note that a mapping can be passed into the integration interface which relates the models of the data provided by the data sources to the global model of the data used by the integrated data source.)

Interface	Functionality
Service Metadata	Retrieve a description of the service, its functionality, available dataset(s) and their descriptions
Registration	Store a description of a service, its functionality, available dataset(s) and their descriptions
Integration	Enables the creation of an integrated dataset from one or more existing datasets
Query (WS-DAI)	Enables a declarative query, in a supported query language, to be posed against the available data
Data Access (WS-DAI)	Supports iterating over a dataset, which may have been generated in response to a query
Subscription (WS-N)	Supports requests to be sent notifications of new data items
Notification (WS-N)	Supports the receiving of notification messages

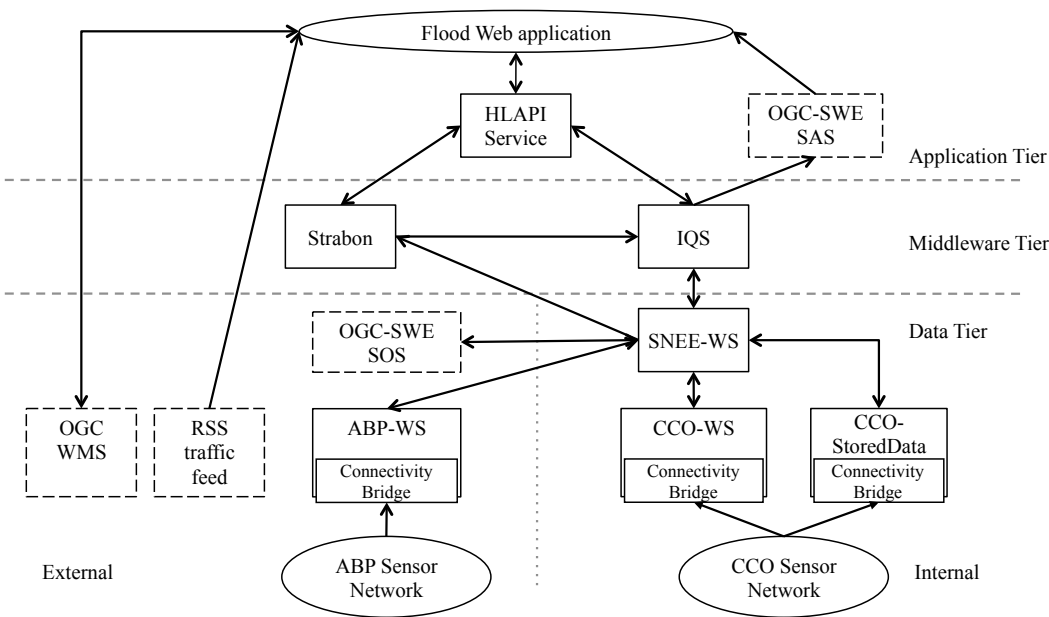
To ease the presentation of the architecture, we draw examples from an environmental decision support scenario for a coastal flood surge. The semantic sensor web service instances deployed for this scenario are given in Table 2. The interactions of these service instances, together with sample OGC-SWE services, are depicted in Figure 3. The scenario consists of data being published by a variety of sources: some using the services defined as part of the semantic sensor web architecture. For example, the CCO-WS service publishes the sensor readings generated by the CCO sensor network [45]; the ABP-WS service publishes sensor readings about tides, waves, and weather conditions generated by a sensor network deployed by the Associated British Ports authority [58]; and the CCO-Stored service publishes

stored information relating to the CCO sensors such as their associate storm threshold value. Others are OGC defined services (e.g., Web Feature Service, SOS, and SAS), while others are *ad hoc* web feeds (e.g., traffic report RSS feed). Note that the instance of the architecture deployed for the flooding scenario makes extensive use of the ontology network presented in Figure 1. However, the architecture does not consider the problem of ontology maintenance.

Table 2. Details of the deployed semantic sensor web services: the service name, the type of service, the interfaces supported, and the endpoint reference. The full list of active service endpoint references is available from <http://www.sensorgrid4env.eu/index.php/services-applications> (URL accessed 24 June 2011).

Name	Service Type	Interfaces	Endpoint Reference
Strabon	Semantic Registry Service	Service Metadata, Registration, Query, Data Access	http://www.sensorgrid4env.eu/services/registry-service
CCO-WS	Streaming Data Service	Service Metadata, Data Access, Subscription	http://www.sensorgrid4env.eu/services/cco-stream
ABP-WS	Streaming Data Service	Service Metadata, Data Access, Subscription	http://www.sensorgrid4env.eu/services/abp
SNEE-WS	Streaming Data Service	Service Metadata, Integration, Query, Data Access, Subscription	http://www.sensorgrid4env.eu/services/snee-stream
CCO-Stored	Stored Data Service	Service Metadata, Query, Data Access	http://www.sensorgrid4env.eu/services/cco-stored
IQS	Semantic Integrator Service	Service Metadata, Integration, Query, Data Access, Subscription	http://www.sensorgrid4env.eu/services/semantic-integrator
HLAPI	Application Service	REST interface supporting HTTP GET	http://id.sensorgrid.ecs.soton.ac.uk/sensors/all

Figure 3. Deployment of semantic sensor web service instances to satisfy the flood surge scenario. Registration interactions have been omitted for clarity of presentation. All services register with the Strabon service, a registry service instance.



The following sections describe the principal capabilities supported by the semantic sensor web architecture and provide details of the services which provide the functionality. Full details of the interfaces and services defined by the architecture are available in [59].

4.1. Registering and Discovering Datasets

The semantic registry service supports the registration of services and their datasets through the *Registration* interface. The deployed registry service, Strabon (Table 2), accepts RDF documents describing the interfaces supported by the service, and the spatiotemporal and thematic description of the datasets. These semantic descriptions are expressed with respect to the network of ontologies [46] depicted in Figure 1. For example, descriptions of sensor networks and corresponding observations refer to the Semantic Sensor Network (SSN) ontology [47]. Note that the registry service implementation is not limited to using terms from the ontology network.

The semantic registry supports the discovery of datasets by answering queries posed through the *Query* interface. The service implementation accepts queries expressed in stSPARQL: a SPARQL extension which has additional support for spatiotemporal constraints [60]. This enables dataset and service discovery based on the spatiotemporal and thematic coverage of the data, expressed in terms of concepts from the ontology network, rather than simply in terms of the functionality offered by the service. For example, an emergency response planner who is responsible for a specific region, say the Solent—the strait of water separating the Isle of Wight from the south coast of England—can specify their region using the relevant concept from the ontology network, viz. `AdditionalRegions:Solent`. The ontology concept contains a definition of the Solent region as a polygon represented using the Well-Known Text (WKT) format [61]. Alternatively, an environmental scientist who focuses on an area of coastline that exhibits particular features and which spans different regions can specify the exact region directly.

The queries posed to the registry service can request any information that is stored in the registry. The information returned is either in the form of SPARQL bound variables or an RDF document, depending on the query posed. Thus, the client can retrieve only the endpoints of selected services or all the information stored in the registry about a service. Figure 4 shows an example query posed to the Strabon registry service to discover the endpoint reference and service type of all data sources for the Solent that provide data about wave height sensor readings. Lines 13 and 14 declare the area covered by the dataset, while line 15 declares the region of interest to be the Solent. The `FILTER` on line 16 ensures that the dataset covers the Solent region. The results are returned as SPARQL bound variables.

Discussion

The functionality offered by the semantic registry service goes beyond that offered by OGC-SWE. An OGC-SWE service may use the OpenGIS catalogue services specification [25] for discovery purposes. The common query language developed for interoperability in this specification is based on an attribute-value pairs data model and attribute-operator-value atomic expressions in a Boolean query. In our approach we are more permissive, in the sense that the architecture allows any query language to be used. The architecture is agnostic to the actual language used by the registry instance, which allows for implementations that use stSPARQL [60], GeosPARQL [62], or SPARQL-ST [63]. In the implemented Strabon registry service [64] the user can use stSPARQL, which is a declarative query language. The queries are evaluated over the stored metadata using thematic and spatial criteria instead of being restricted to attribute-operator-value methods that OGC-SWE provides. This is more general in principle. Also, the concrete realisation of the semantic registry service by any RDF store offers a semantics-based data

model which is more expressive than an attribute-value pair data model. Additionally, in our approach the user can use real-world concepts, e.g., semantic terms that are defined in a dataset exposed as linked data such as LinkedGeoData [65], to refer to spatial geometries besides using coordinates to explicitly specify them.

Figure 4. Declarative query to the Strabon registry service in the stSPARQL language. The query discovers service endpoints and their type for data sources providing wave height measurements for the Solent region.

```

1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX Services: <http://www.sensorgrid4env.eu/ontologies/ServiceOntology.owl#>
3  PREFIX CoastalDefences: <http://www.sensorgrid4env.eu/ontologies/CoastalDefences.owl#>
4  PREFIX AdditionalRegions: <http://www.ordnancesurvey.co.uk/ontology/v1/AdditionalRegions#>
5
6  SELECT DISTINCT ?endpoint ?servicetype
7  WHERE {
8      ?service rdf:type Services:WebService;
9              Services:hasEndpointReference ?endpoint;
10             Services:hasServiceType ?servicetype;
11             Services:hasDataset ?dataset.
12      ?dataset Services:includesPropertyType CoastalDefences:WaveHeight;
13             Services:coversRegion ?region.
14      ?region Services:hasSpatialExtent ?geo1.
15      AdditionalRegions:Solent Services:hasSpatialExtent ?geo2.
16      FILTER(?geo1 COVERS ?geo2).
17  }

```

GeosPARQL is being developed in an ongoing OGC standard working group [66] for representing and querying geospatial data expressed in RDF. Earlier contributions in this area include [63,67,68] and our original paper on stSPARQL [60]. stSPARQL [60,64] has many commonalities with the recent OGC work on GeosPARQL. Both approaches represent geometric objects as spatial literals and may be encoded in various formats like GML, KML, or WKT. In both approaches, basic spatial functions, spatial predicates, and functions that support spatial analysis, are mapped to extension functions that may be used in the SELECT clause or the FILTER conditions of a query. In GeosPARQL, basic spatial functions and spatial predicates are also mapped to RDF properties. This allows GeosPARQL to be easily combined with other representational frameworks that formalise qualitative spatial reasoning (e.g., rules as in [69]). A point where the two languages differ is that GeosPARQL offers a minimal ontology that can be used for representing features and geometric objects. In our approach, we impose very minimal requirements to semantic web developers; all they have to do is utilise a new literal datatype and they are free to define and use their own spatial ontologies. The approach followed by [70] is also the same, although the authors concentrate more on indexing and query processing and less on defining a complete geospatial extension of SPARQL.

4.2. Publishing Datasets

The data source component depicted in Figure 2 denotes three distinct types of data service: (1) streaming data services that enable access to sensor data as continuously updated streams of values; (2) stored data services that enable access to data stored in repositories, such as a relational database or a triple store, including sensor deployments that load their readings directly into a database; and

(3) external data services that enable access to data via some defined service interface, e.g., an OGC Web Map Service for publishing map images (OGC-WMS [71]), or an OGC-SWE SOS [15].

4.2.1. Streaming Data Service

A streaming data service is defined as an extension to the Open Grid Forum (OGF) web service recommendation for data access and integration (WS-DAI) [72]. Such a service provides the *Service Metadata* interface to enable clients to discover metadata about the service and the datasets that it makes available. The metadata is returned as an XML document that conforms to the WS-DAI standard and has been extended to contain an RDF description of the service and its datasets, using terms from the network of ontologies depicted in Figure 1. Registering a streaming data service simply involves sending the metadata document, containing the RDF description, to the *Registration* interface of the registry service.

In its most basic form, the streaming data service makes the readings from a sensor network available as a stream of data values through one, or both, of the *Data Access* or *Subscription* interfaces. These enable clients to access the readings in the time series as they become available. In the case of the *Data Access* interface, the client must poll for the value. The rate at which the client polls for new values can be configured based on the information contained in the metadata, e.g., the expected rate of the data stream. In the case of the *Subscription* interface, the client must expose the *Notification* interface for receiving notification messages that new sensor readings have been generated and published. The publish/subscribe mechanism uses the OASIS Web Services Base Notification (WS-N) standard [8].

As a specific example, consider the sensor network deployed by the Channel Coastal Observatory (CCO) [45] around the coast of England and published through the CCO-WS streaming data service (Table 2). This is an active wireless sensor network consisting of 43 sensor nodes deployed at strategic locations, many along the south coast of England. Each node in the network generates its own stream of data values named after the location of the sensor node. There are three distinct types of stream measuring properties of waves, tides, or meteorological conditions; an example of the schema of each type of stream is shown in Figure 5. There are 24 nodes measuring wave properties with the same schema as *envdata_haylingisland*, 7 nodes measuring the properties of tides with the same schema as *envdata_sandownpier_tide*, and 12 nodes measuring meteorological values with the same schema as *envdata_lymington_met*. Since the sensor nodes in the CCO sensor network have a fixed sensing period, viz. they take a new reading every 10 minutes, the rate of the streams is published in the metadata document. This enables clients to configure the rate at which they poll for data.

Streaming data services can also provide a *Query* interface. The *Query* interface supports clients in providing a declarative description of their data needs as a query. The *Query* interface does not specify the language in which the query is expressed; this is available in the service metadata. This allows services to support a query language suitable for their data, e.g., SNEEql for continuous queries over streaming relational data [29]. The response to a query operation is the generation of a data stream whose values satisfy the query. This new stream is made available either through the *Data Access* or *Subscription* interface. Finally, the service can also offer the *Integration* interface to support queries involving more than one data source, e.g., another streaming data service. In the case that an external data source is used, the *Notification* interface can be provided to support subscriptions to external sources. An example of this more feature-rich streaming data service is made available by the SNEE-WS service

(Table 2), which uses the SNEE query engine to process streaming data queries [28]. The deployed SNEE-WS service is configured to use the streams published by the CCO streaming data service as its data source. Thus, it provides a query interface to that data. This enables the integration service (Section 4.3) to pose queries over the streams published by the CCO-WS, e.g., the 24 wave streams can be merged into a single wave stream using a UNION query. SNEE-WS is also capable of evaluating queries over a mix of streaming and stored data services with a well-defined evaluation semantics (details of a motivating case for such queries are given in Section 4.3).

Figure 5. Example schema of the streams generated by the Channel Coastal Observatory wireless sensor network deployment; published as a streaming data service (CCO-WS).

```
envdata_haylingisland(ts:timestamp, lat:float, lon:float,
                    HMax:float, Tp:float, Dirp:float, Sprp:float,
                    Tz:float, Hs:float, TSea:float)

envdata_sandownpier_tide(ts:timestamp, Observed:float, Tz:float,
                        Hs:float, HMax:float, Tp:float)

envdata_lymington_met(ts:timestamp, TAir:float, WDir:float,
                     GustSpeed:float, WindSpeed:float, AirPressure:float)
```

4.2.2. Stored Data Service

The stored data service enables access to data stored in repositories such as relational databases, XML stores, or triple stores. This can include retrieval of values from a sensor network source which are first stored in a database, access to the historic values from the data stream, or data associated with the deployment of the sensor network, e.g., the location of the sensors or associated threshold values. We have adopted the OGF WS-DAI [7,72] recommendation for the stored data service and the reference implementation based on OGSA-DAI [73].

Clients can discover the metadata associated with the stored data service through the *Service Metadata* interface, which returns an XML document describing the service and the datasets. The stored data service enables declarative queries to be evaluated over its data content through the *Query* interface. Since the answer to a query can be potentially large, the answer can be broken into chunks that are retrieved through the *Data Access* interface.

The metadata associated with the CCO sensor network and the history of sensor readings is published as a database through the CCO-Stored stored data service (Table 2). One of the relations published in this database associates a storm threshold value with each of the sensor data streams and has the schema

```
locations(id:int, latitude:decimal, longitude:decimal,
         storm_threshold:decimal, deployed:timestamp).
```

Section 4.3 will show how the stored storm threshold data can be combined with the live sensor data to detect an over-topping event.

4.2.3. External Data Services

The specification of the streaming and stored data services are targeted at specific types of data, *i.e.*, time series generated by a sensor network and data stored in a database, respectively. For other types

of data, e.g., map features and bitmap images, there are existing standards such as the OGC Web Feature Service (OGC-WFS) [74] and the OGC Web Map Service (OGC-WMS) [71] that are suitable. These can be used in our conception of a semantic sensor web by manually registering a semantic description of the service and its data with the registry service. A web page interface to the Strabon registry service has been made available for this purpose (<http://www.sensorgrid4env.eu/services/registry-service/Store>). Indeed, the use of OGC services is the approach adopted with the output of the environmental models for forecasting sea-state, which make available a map image of the raster grid as an OGC-WMS (see Section 5.1). Our approach to external data services also supports the use of other sources of time series data such as made available by an OGC-SWE Sensor Observation Service [9]. Note that these must also be manually registered in the registry.

4.2.4. Discussion

The functionality offered by the streaming data service combines the data publishing functionality offered by the OGC-SWE SOS and SAS services, whilst offering richer data processing and filtering capabilities. The SOS only supports clients polling for data while the SAS supports the pushing of data to clients that have subscribed. Both forms of interaction are supported by the streaming data service. By exposing the *Data Access* interface, the streaming data service supports clients in polling for sensor readings. By exposing the *Subscription* interface, the streaming data service is able to push sensor readings to subscribers in a timely fashion. The streaming and stored data services do not prescribe a data model or format for the data published, unlike the OGC-SWE services which requires that data conforms to the O&M syntactic model. This provides a lower entry requirement for data publishers as they only need to describe their data, not map or transform it to some common data model that may not accurately capture their data. Finally, the streaming and stored data services offer the possibility of supporting declarative queries by exposing the *Query* interface. A declarative query language, such as SNEEql [29], provides the possibility to correlate stream values, both within the stream and across streaming and stored sources, and compute aggregates such as an average. This goes beyond the simple attribute-operator-value filtering offered by the SOS.

4.3. Data Fusion: Integrating Heterogeneous Datasets

A key requirement for a sensor web is the ability to combine data from multiple sources. Existing approaches rely on either the end user manually combining datasets in a spreadsheet, or using data fusion or mash-up approaches which simply juxtapose datasets from multiple sources. The latter cases rely on the end user to visually interpret the result and the resulting data set has no coherent semantics. Such an approach can result in useful data but can be more laborious.

A key ability of the semantic sensor web is integrating and correlating data from heterogeneous sources: both in terms of the modality of the data (*i.e.*, streaming and stored), and the representation of the data (*i.e.*, the schema used). Through the *Integration* interface, the integration and query service (IQS Table 2) supports the creation of a *virtual* data source in which the data from multiple physical data sources appear to co-exist in a single data model. The virtual data source can be queried through the *Query* interface and query answers retrieved through either the *Data Access* or *Subscription* interfaces.

The IQS is configured to expose an ontological view over a set of data sources through a mapping document. The mapping document relates the data conceptualisations exposed by the data sources to a global data model. The mappings are expressed in terms of selections and transformations over the data sources, and can be created either manually or with the help of a mapping tool. For example, the IQS service deployed for the flooding application exposes the data published by the CCO-WS service and the ABP-WS service using the *SSN* ontology. Users and applications can pose queries expressed in SPARQL_{Stream} over the ontology. These are transformed, according the relationships expressed in the mapping document, by the IQS into queries over the data sources. The generated queries are then passed to the SNEE-WS service with the answer streams flowing back. Full details of the mappings and the semantic integrator implementation can be found in [30].

A motivating use case for the integrator, drawn from the flood surge application, is to expose sensor readings and associated metadata for the south coast of England. A user, most likely a domain expert, creates the virtual source by providing a mapping document to the IQS through the *Integration* interface which states how to relate the source streams of the CCO-WS and the ABP-WS together with the tables of the CCO-Store to the ontological model of the *SSN*. The integrated data source is automatically registered with the registry service, making it available to all users. Any user can then pose queries over the resulting data source to explore aspects of the data, or indeed be informed of interesting events as characterised by queries over the integrated source. For example, it is desirable to be informed when a sea defence has been over-topped by a wave. This can be determined by a query that relates the wave height readings from the sensor network with the associated threshold value stored in the database. The query over the integrated observation model is shown in Figure 6 expressed in SPARQL_{Stream}. Line 16 of the query relates the data stored in the CCO-Store database with the current sensor reading from the CCO-WS, the keyword 'NOW' on line 6 limits the sensor readings to the current time-point. The query is evaluated by posing a translated query to the SNEE-WS distributed query processing service. The SNEE-WS service poses a query to the *locations* table of the CCO-Store data service and correlates the results with the streams consumed from the CCO-WS and ABP-WS.

Note that the new data stream generated by the query in Figure 6 can be used to inform users whenever such an over-topping event has taken place. For example, it can be used as an input feed to an alert service such as the OGC-SWE sensor alert service [9], as depicted in Figure 3.

Discussion

The OGC-SWE architecture does not have a service specifically aimed at integrating data from multiple data sources. In the SANY project, a Cascading Sensor Observation Service [24] was defined and implemented to support the following use cases: (i) to merge streams and republish the result; (ii) to wrap legacy data sources so that they can be accessed through the SOS interface; (iii) to compute simple transformations on the data, e.g., unit conversion or calculating moving averages; (iv) to perform load balancing across deployed services; and (v) to replicate data. The first three use cases are covered by the semantic integration service, except that the semantic integration service can also perform transformations between different views of the data, *i.e.*, from the many views of the data publishers to the integrated ontological view exposed by the integration service. Use cases (iv) and (v) are deployment issues rather than architectural decisions. A deployment of the semantic integration service may choose to replicate

its service deployment across several machines and it may choose to replicate the source data or fetch it on-the-fly.

Figure 6. Declarative query in the SPARQL_{Stream} language which characterises an over-topping event over an ontological observation model for sea-state readings. The event is characterised by the measured wave height being greater than the associated storm threshold value for a specific sensor.

```

1  PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
2  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  PREFIX CoastalDefences: <http://www.sensorgrid4env.eu/ontologies/CoastalDefences.owl#>
4
5  SELECT ?location ?waveHeight ?stormThreshold ?observedTime
6  FROM NAMED STREAM <http://sensorgrid4env.eu/ns#waveStream.srdf>[NOW]
7  WHERE {
8      ?WaveObs rdf:type ssn:Observation;
9              ssn:observationResult ?waveHeight;
10             ssn:observationResultTime ?observedTime;
11             ssn:observedProperty ?waveProperty;
12             ssn:featureOfInterest ?defence.
13     ?defence CoastalDefences:hasLocation ?location;
14             CoastalDefences:hasStormThreshold ?stormThreshold.
15     ?waveProperty rdf:type CoastalDefences:WaveHeight.
16     FILTER(?waveHeight > ?stormThreshold).
17 }
```

4.4. HLAPI: Observation Linked Data Service

The previous subsections have presented the core web services that form a service-oriented architecture, specifically those which comprise the data and middleware tiers of the semantic sensor web architecture (Figure 2). The focus of those services is to provide programmatic access to the data for the creation of orchestrations to enable discovery, publication, and integration of data. The top tier of the semantic sensor web architecture provides services which are domain-specific and user-facing. An example of such a service is the High-Level Application Programming Interface (HLAPI) service (Table 2). The focus of the HLAPI service is to support higher-level tools, particularly web-based applications and mashups. For example, to enable time series sensor data to be presented in a mash-up together with other contextual datasets. This is achieved through a linked data [75] approach to publishing sensor observations, received from the data services and semantic integrator, in combination with REST conformant representations specific to the geographic and environmental domain.

While it is possible to send and receive SOAP messages from code running in a web browser using a library such as CXF [76], this approach is, in general, not followed in mash-up development. Web application developers typically expect to retrieve data through a REST interface [21] supporting the four HTTP operations, viz. GET, POST, PUT, and DELETE, that are supported in the execution environment provided by a web browser; services following this pattern are deemed “RESTful”. To make full use of the semantic content provided through the semantic sensor web while exposing data in a manner consistent with REST design, it is also desirable for an interface to follow the principles of linked data [75]. That is: (1) use URIs as identifiers for resources; (2) use HTTP URIs so that resources can be looked up; (3) provide meaningful standards-based representations of the resources when they are looked up containing links to

more data—also known as the follow-your-nose principle; and (4) include URI links to related resources to enable discovery through link navigation.

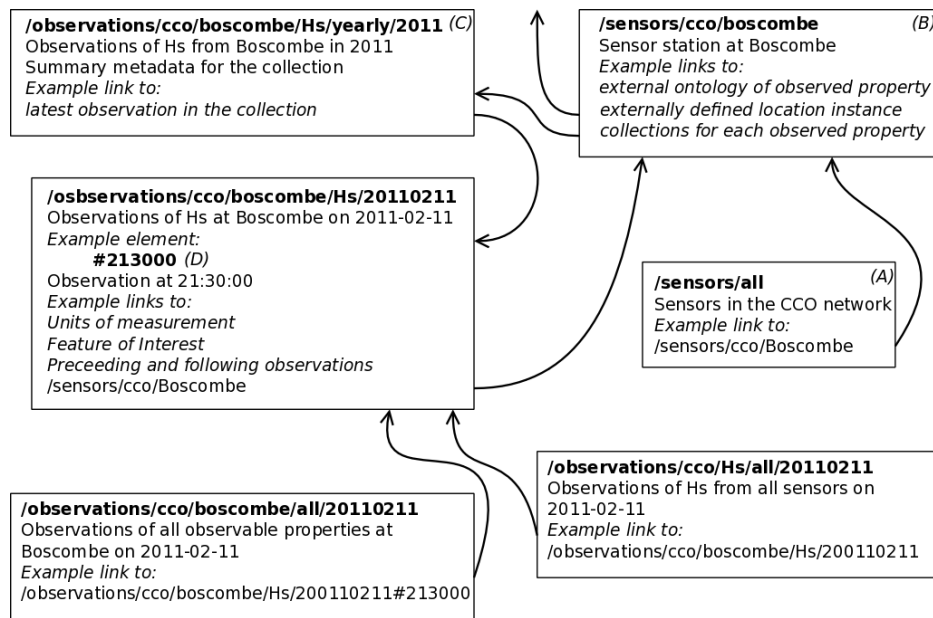
In the environmental area, many mash-ups and web applications are further characterised by an approach by which datasets are overlaid on a base map with each dataset displayed being transformed into a layer (as depicted in Figure 10 and explained in Section 5.2). To create these map layers, the web application needs to access the data through an interface that is supported in the restricted execution environment provided by a web browser, and in a format that can be converted to a layer in that restricted execution environment (e.g., a GeoJSON array, or WFS GML). These formats are, when accessed through a REST interface, additional representations that complement those provided as linked data (viz. RDF).

Applying these complementary REST and linked data approaches over the ontology network described in Section 3, a High-Level API (HLAPI) for sensor observation data has been developed [35,77,78]. It enables web applications to access and navigate the data (practising the follow-your-nose principle) using resources comprising the observations, sensors, and time-series collections of observations grouped by the property being observed, the gathering sensor, and temporal boundaries. Links are made and maintained between collections and observations—a single observation is likely to be included in several collections, and a collection may in turn be a constituent of several higher-level collections—and to externally defined linked data sources, e.g., relating sensor positions to geopolitical boundaries, or observed properties to their definitions in the domain. Representations include those needed for linked data applications (viz. RDF, also via SPARQL), for use within geographic applications and for layer generation (viz. O&M GML, GeoJSON, WFS GML) and for general human-readable browsing (viz. HTML). The HLAPI is implemented as a service which generates the structure of resources and links between related data resources according to the rules provided in the service configuration. This has two main parts: the alignment of the incoming data streams with a single data model, which is directly inherited from the integrator service (Section 4.3) when that is the providing data service; and the declaration of an API such that resources are appropriate to the specific data being published, e.g., defining collections of manageable size derived from observation frequency.

For example, to support mash-up developers in the reuse of sensor data published by the CCO sensor network (Section 4.2), we have instantiated a HLAPI service to provide access to the observations published by the virtual data source available from the integration service. The deployed HLAPI service automatically generates URIs for sensors and each observation as data is received from the integration service. It also dynamically organises, and creates as required, the collections to group observations according to their location, the property that is measured, and the time at which it was measured. A selection of resources from this example deployment are shown in Figure 7. A semantic web client might initially access the HLAPI service using the URI for all sensor resources (A) (also Table 2). The client can follow the link to the Boscombe sensor resource (B) and request an RDF representation of the Boscombe sensor resource. This representation would contain a semantic description of the sensor in terms defined by the ontology network: including its capabilities, location, and annotated links to collections to which it has contributed observations, such as an annual collection of wave height measurement (C). The latest observation, say 9.30 pm on the 11th February 2011 (D) is accessible through a link. Alternatively a client may receive a link to a specific observation, such as (D). From this resource, the client can follow links to preceding or following observations of wave height from the Boscombe sensor, or to collections

of which the observation is a constituent. The RDF representation is best used for semantically navigating the dataset due to library and tooling support. However, once the desired resource is located, the client can request any representation, e.g., GeoJSON to create a layer, through content negotiation.

Figure 7. A selection of resources and links provided by the CCO deployment of the HLAPI (not comprehensive).



Discussion

Beyond an early prototype of the API design outlined in [35], there have been multiple proposals for exposing sensor related information as linked data, each with differing motivations and foci: automated conversion from OGC standards and services [36], alignment with foundation ontologies [79], publishing linked sensor locations and attributes [80], sensor discovery over linked data [81], and integration from multiple sensor sources into a single service [82].

While the HLAPI touches upon the issues raised in several of these works, the primary focus during its development was instead the design and deployment of APIs that are accessible and relevant to a developer working in the domain, on linking the semantics of the domain to observations so they can be reused in web applications and mashups, and on utilising the semantics of the domain model to simplify the configuration and deployment of services.

The SANY architecture [20] outlines an abstract resource model which can be used to align OGC-SWE services with REST practises, most recently leading to a proposal for a RESTful interface to SOS services [83]. The HLAPI, while taking a similar approach to RESTful publication of resources and representations, internally uses an RDF model as its primary representation rather than the provision of resources through an extension of OGC-SWE services. This enables the HLAPI to preserve and publish resources based upon the semantic structures provided by the semantic sensor web architecture, and to exercise semantic mappings, using the same domain ontologies, to simplify configuration of HLAPI deployment.

5. Interacting with the Semantic Sensor Web

This section presents the interactions of applications with the semantic sensor web architecture presented in Section 4. We first consider the generation of flood warnings based on the readings from the sensor networks and other sources of data. The approach uses environmental modelling techniques to predict the changes in sea-state for a period of time in the future based on the most recent sensor readings. The results are themselves published as (external) data sources in the deployed semantic sensor web. We then describe a web application that has been created using the services and functionality provided by the semantic sensor web.

5.1. Forecasting Flooding Events

Environmental models that forecast the behaviour of the sea-state have the potential to predict future flooding events (within given error margins): specifically where, when, and how severe they may be. These models are computationally expensive to run and depend on data from a wide variety of data sources including sensor data, weather feeds, and data stored in databases. In the context of forecasting flooding events and planning a response, three environmental model services have been developed. These models forecast the sea-state, likelihood of over-topping or breaching flood defences, and the water levels on the flood plains. The output of the models are exposed to applications using the semantic sensor web as layers published by OGC web mapping services (OGC-WMS) [71] which have their semantic descriptions registered with the registry service.

The first service models the mean wave height and sea level for the next eight hours. The model uses sea-state data from the CCO sensor network together with meteorological data as input. The result is a finite element mesh which predicts the sea-state (level and wave heights) at each point on the mesh and is represented as a raster grid. The raster grid output format (GeoTIFF) is exposed as a collection of layers distinguishable by their order of precedence and a timestamp through an OGC-WMS.

The second modelling service uses the output of the sea-state forecast model together with information stored about sea defences in databases such as the National Flood and Coastal Defence Database (NFCDD) of the UK Environment Agency [3]. The input data is processed using an empirical over-topping formula [84] to predict the probabilities of sea defences being over-topped. The output of the model is a collection of ESRI ASCII Grid files which can be served by an OGC-WMS.

The final modelling service predicts the depth of water on the flood plains in the event of an over-topping event. It runs the LISFLOOD-FP inundation model [85] using the tidal water levels provided by the sea-state modelling service. The output is a raster grid showing the maximum forecast flood level for each cell at each time point. The result is again made available as map layers through an OGC-WMS.

5.2. Emergency Response Web Application

Tidal surges and the resulting flooding have the potential for devastating effects to businesses and lives. A wide range of users, including emergency response planners, harbour masters, and members of the public, need to be supported in interacting with the data in these situations. We have developed a web based application, *i.e.*, one that executes in the limited environment offered by a web browser and only permitted to make HTTP calls, which enables users to discover data sources based on their spatiotemporal

and thematic content, and to juxtapose that content as layers on a map. Screenshots from the application are shown in Figures 8, 9, 10, and 11, and will be discussed in the remainder of this section.

Figure 8. Screenshot of the login screen for the web application.

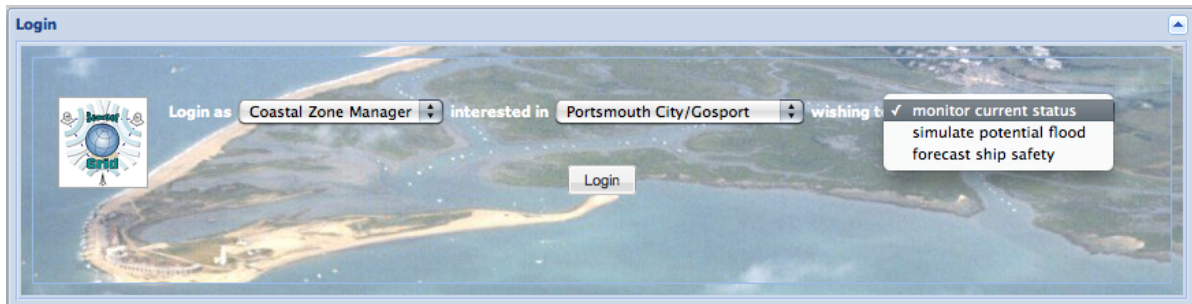
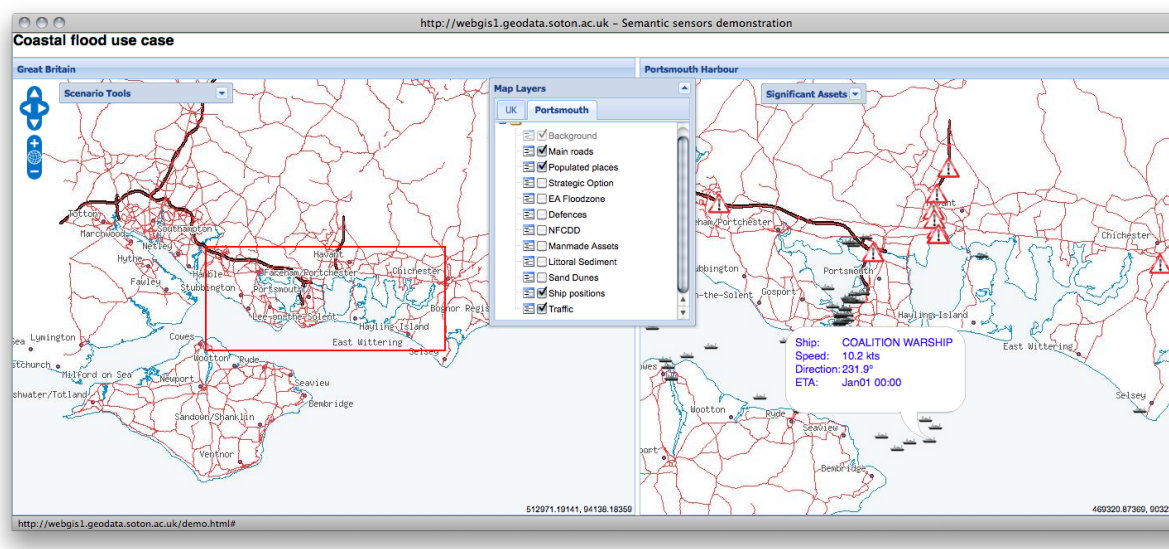


Figure 9. Screenshot of the initial layers displayed for the role *Port Authority* the region *Portsmouth City/Gosport*, and the task of *Forecast ship safety*.



The application exploits the functionality provided by the services of the presented semantic sensor web: specifically the ability to discover datasets and services, and to orchestrate the services to integrate heterogeneous data sources and dynamically publish the resulting dataset through a REST interface. For example, the application can interact with data in native web formats, such as GeoJSON, by using an appropriately configured HLAPI service instance (Section 4.4) to return observations through a REST interface. In turn, the HLAPI service instance can use the semantic integration service (Section 4.3) to retrieve data stemming from multiple heterogeneous data sources, both in terms of modality and data model, as a unified virtual data source which presents its data as instances of an ontology.

In this section we describe the web application and the issues faced in its development. The application is an exemplar of the type of mash-up that can be developed on top of the semantic sensor web. The web application is available from <http://www.semsorgrid4env.eu/services/dynamic-demo> and <http://www.semsorgrid4env.eu/services/static-demo>. The latter provides a more stable demonstration interface

highlighting stable functionality, while the former aims to be as dynamic as possible, at the potential expense of stability/usability.

Figure 10. Screenshot showing layers generated from sensor readings.

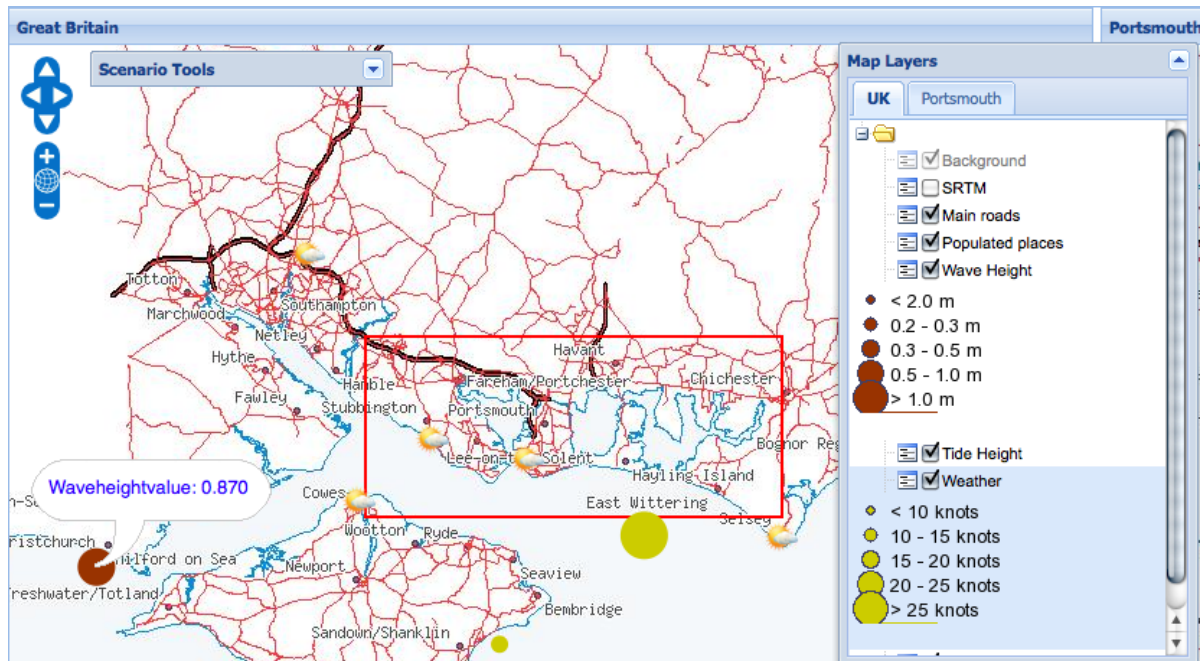
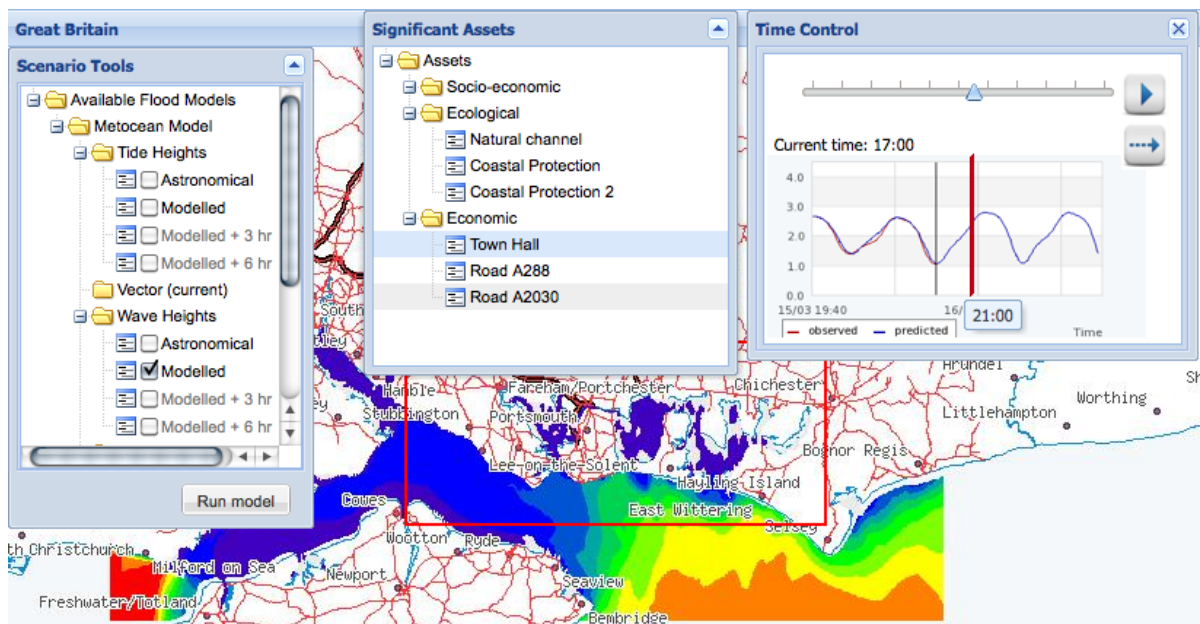


Figure 11. Screenshot displaying the output of the wave height flood model (Section 5.1).



5.2.1. The Solent and its Users

The Solent region is the area surrounding the Isle of Wight off the south coast of England. This region has a complex tidal and wave pattern which generates a demand for sea-state forecasts. It is a busy shipping area servicing the ports of Southampton and Portsmouth with passenger ferries, cargo

ships, and military vessels as well as a large number of recreational water users. The coastline comprises a variety of built up areas such as the cities of Southampton and Portsmouth, and is home to a number of sites with statutory protection, including special areas of conservation. As a consequence, there is a wide variety in the types of users who are interested in interacting with the data about the region made available by the semantic sensor web. Table 3 illustrates the types of user groups and their interests.

Table 3. User groups for the flood web application.

User Group	Role
Coastal Zone Manager	Management of environmental quality, flood risk management strategy, preparation and response and infrastructure management.
Flood Modeller	Development of scenarios of hazards and risk. Generation of wave over-topping, flood envelopes and prediction of potential assets at risk.
Emergency Services	Response to assets/population at risk, early warning and alert services, defence rescue and evacuation.
Ports Authority	Management of port operations, pilotage services and risk management.
General Public	Interest in potential flooding events and how they will affect them.

5.2.2. Application Characteristics

The estuarine surge and coastal flooding emergency response web application is built on top of the semantic sensor web architecture using the open source mash-up development tool ExtJS [86] and its geospatial interface extension [87]. The web application is based around the standard GIS approach of displaying data contextually as layers on a map. In order to generate these layers we make use of the OpenLayers toolkit [88]. OpenLayers provides a JavaScript API that can be executed within the web browser execution environment, and by which data can be converted into layers for displaying on a map.

The web application aims to support a variety of users, from professional emergency response planners to members of the public, in gaining vital information about a flooding event. This requires dynamically retrieving a variety of data sources and supporting user interaction with these. The typical interaction model is first to discover relevant sources of data based on the spatiotemporal and thematic coverage by querying the registry service (Section 4.1). The user can then select which sources should be added to their display. This can involve using data directly from the original data sources (e.g., the data resulting from the flood models in Section 5.1), using a processed version of the data which is appropriately presented for inclusion in a web application (e.g., data represented as a GeoJSON array from the HLAPI service described in Section 4.4), or an integrated source of data which combines the data from a variety of data sources (*i.e.*, data from the integration service described in Section 4.3).

In the rest of this section, we describe the interactions of the application with the semantic sensor web architecture by following a use case for the port authority user (Table 3).

5.2.3. Application Description

Users access the web application through a login screen, shown in Figure 8, which allows them to select their role, the region they are responsible for, and the task that they wish to conduct. These are presented as options in drop-down selection boxes which are populated with terms from the ontology network [46]. For example, the choice of role comes from the concepts in the *Role* ontology. This

provides an initial characterisation of the data that is relevant for the interaction, *i.e.*, the values provided parameterise the queries sent to the registry in order to discover relevant data sources. For the example of the Queen's Harbour Master at Portsmouth, they would select their role as *Port Authority*, their region as *Portsmouth City/Gosport*, and their tasks as *Forecast ship safety*. These options parameterise the query sent to the registry service to discover relevant data sources based on the spatiotemporal and thematic coverage.

The result of the login process is a screen presenting the user with two map views side-by-side (shown in Figure 9), based on the region selected, *viz.* *Portsmouth City/Gosport*. The left pane displays a zoomed-out map providing context while the right pane displays a zoomed-in map on the region selected. Both maps are superimposed with layers presenting data from a variety of sources that satisfy the queries sent to the registry. The available layers are shown in the *Map Layers* pop-up window, from which the user can select the layers that they wish to be displayed on each map. Initially the context map in the left pane shows the populated areas and the main roads. The detailed map in the right pane includes the same background layers and also displays two additional layers providing details of shipping in the region and traffic alerts respectively. The shipping information is displayed, with a ship icon indicating the position of each vessel, from a streaming data source publishing data obtained from the automatic identification system (AIS) network used to track ship positions [89]. The user can discover details of the vessel name, heading, and speed in a pop-up balloon by hovering their cursor over the icon. Similarly, the traffic alerts, shown as warning triangles, are displayed from an external RSS web feed.

The *Map Layers* pop-up window shows all possible layers that can be chosen for display, *i.e.*, those that were discovered by the queries to the registry service. The majority of the layers present information located at a specific point in space, *i.e.*, sensor readings. The screenshot in Figure 10 shows a pop-up balloon displaying the latest wave height reading from a node in the CCO sensor network. This value is provided by an instance of the HLAPI service (Section 4.4) which in turn obtains it from the sensor data stream produced by a query to the virtual data resource published by the integration service (Section 4.3). The interpretation of the size and colour of the circles displaying sensor readings are provided in the *Map Layers* window. Other layers that can be selected display weather readings coming from the deployed sensor network (published through the CCO-WS and displayed as yellow circles), forecasts from public meteorological web services (displayed as weather symbols), and details of flood defences coming from stored data sources such as the UK Environment Agency's National Flood and Coastal Defences database (NFCDD [3], not shown).

The screenshot in Figure 11 shows a model layer presenting the forecast wave heights as generated by the service described in Section 5.1. The result of the model is superimposed on the sea with different colours (shading) depicting regions with different wave heights. The choice of model to display is controlled by the *Scenario Tools* pop-up window and the time point in the model is controlled by the *Time Control* pop-up window. Users can select an appropriate model and forecast period (Astronomical, Modelled nowcast, +3 hours, +6 hours) and then use the time control to animate through 24 hours of the selected model, comparing the model visualisation with a display of the current tide cycle at that time, shown within the *Time Control* pop-up. The time control is also linked to the *Significant Assets* window which is populated with the list of man-made assets (road or road sections, buildings, etc.) which are potentially endangered by a flooding event at the period of time defined by time control slider. By

selecting different models and forecast periods, decision makers can ultimately compare various flood development scenarios and thus improve risk management procedures.

Development of the flood application is ongoing and forthcoming improvements include a tool for selecting two sources of data, discovered through the semantic registry, and creating a dynamic request to the semantic integration service to combine them, thus giving end users the ability to produce new integrated data sources. The tool is also being demonstrated to, and will be evaluated by, potential users for the Solent region.

6. Conclusions

We have presented a semantic sensor web architecture that comprises a core collection of services that form a service-oriented architecture for data publication, discovery, and integration. The architecture also defines a resource-oriented user-facing web service exposing sensor readings as a linked data resource. The semantic sensor web architecture supports the four capabilities identified in Section 1: (1) identify relevant sources of data; (2) access sensor data in near real-time; (3) correlate disparate heterogeneous sources; and (4) support multiple conceptualisations.

The presented semantic sensor web can be used in conjunction with the services of the Open Geospatial Consortium Sensor Web Enablement (OGC-SWE) [9], as described in Section 4 and depicted in Figure 3: data published through OGC-SWE services can be incorporated into our semantic sensor web and data resulting from the semantic sensor web can feed into higher-level OGC-SWE services. However, the semantic sensor web architecture presented goes beyond OGC-SWE in several important ways. First, it supports data discovery based on semantic descriptions of the services and the datasets that they publish. This enables datasets to be identified based on their content, *i.e.*, their spatiotemporal and thematic coverage, rather than simply matching syntactic strings. Additionally, the relationships between concepts in the ontologies can be exploited for discovering suitable services and datasets. Second, it assists in the integration of heterogeneous datasets from autonomous providers by giving rise to semantically coherent models over the data. This enables the creation of *virtual* datasets that combine data from multiple sources, and the discovery of relationships in the virtual dataset. Note that each set of data may have been published according to a different conceptualisation. The integrator enables these to be related into a coherent conceptualisation. Third, it enables arbitrary models over the data to be used, *i.e.*, the service interfaces are not tied to the model used. The service interfaces are generic, *i.e.*, the operations supported, and their parameters, are not defined in terms of the underlying data model or representation. Instead, declarative query languages are used to express the data requirements. We note that version 2 of OGC-SWE is moving in this direction with the definition of the eventing service [31]. This ensures that the semantic sensor web is generic and extensible, and can be used in applications beyond environmental monitoring, *e.g.*, supply chain monitoring or healthcare, without redefining and implementing the services. Finally, the semantic sensor web architecture supports push-based delivery of data in a timely fashion, directly from the data source. Again, we note that OGC-SWE version 2 is following a similar approach of using the Web Services Base Notification standard [8], although it requires a separate service from the data publication service (*viz.* SOS) [31].

An example web application has been presented which uses the presented semantic sensor web to deliver a semantically-enabled and dynamically-constructed environmental decision support and flood

emergency management tool. The tool is self-configuring in so far as it is based on the role and area of interest specified by the user. The tool enables a role-specific dynamic decision support environment to be experienced by the user. Semantic registry queries are used to discover relevant sources of real-time and historic sensor observations, modelled variables including flood forecasts, and other contextual data. New information sources can be created through the integration service by combining historic and real time sensor measurements, model outputs and a variety of spatially-enabled socio-economic variables. In addition to the use of open standards, the exclusive use of open source technologies without recourse to proprietary software makes the methodology particularly suitable for use by decision makers with limited resources and budget constraints and in rapid response to new conditions, e.g., in an emergency response scenario. The application is not specifically tied to the south of England, or to flood response planning, and can be implemented for sensor networks deployed anywhere in the world.

Acknowledgements

This work has been supported by the European Commission project SemSorGrid4Env (FP7-223913). <http://www.sensorgrid4env.eu>

References

1. Nittel, S. A survey of geosensor networks: Advances in dynamic environmental monitoring. *Sensors* **2009**, *9*, 5664–5678.
2. Havlik, D.; Schade, S.; Sabeur, Z.A.; Mazzetti, P.; Watson, K.; Berre, A.J.; Mon, J.L. From sensor to observation web with environmental enablers in the future internet. *Sensors* **2011**, *11*, 3874–3907.
3. National Flood and Coastal Defence Database (NFCDD). Available online: <http://www.scisys.co.uk/casestudies/environment/nfcdd.aspx> (accessed on 18 August 2011).
4. W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. Recommendation, W3C, 2009. Available online: <http://www.w3.org/TR/owl2-overview/> (accessed on 31 August 2011).
5. Manola, F.; Miller, E. *RDF Primer*; Recommendation, W3C, 2004. Available online: <http://www.w3.org/TR/rdf-primer/> (accessed on 31 August 2011).
6. Prud'hommeaux, E.; Seaborne, A. *SPARQL Query Language for RDF*. Recommendation, W3C, 2008. Available online: <http://www.w3.org/TR/rdf-sparql-query/> (accessed on 31 August 2011).
7. Antonioletti, M.; Krause, A.; Paton, N.W.; Eisenbrg, A.; Laws, S.; Malaika, S.; Melton, J.; Pearson, D. The WS-DAI family of specifications for web service data access and integration. *SIGMOD Rec.* **2006**, *35*, 48–55.
8. Graham, S.; Hull, D.; Murray, B. *Web Services Base Notification 1.3 (WS-BaseNotification)*. Standard OASIS, 2006.
9. Botts, M.; Percivall, G.; Reed, C.; Davidson, J. OGC® Sensor Web Enablement: Overview and High Level Architecture. In *Proceedings of the 2nd International Conference on GeoSensor Networks, GSN 2006*, Boston, MA, USA, 1–3 October 2006; Volume 4540, pp. 175–190.

10. Usländer, T.; Jacques, P.; Simonis, I.; Watson, K. Designing environmental software applications based upon an open sensor service architecture. *Environ. Model. Softw.* **2010**, *25*, 977–987.
11. Delin, K.A. The sensor web: A macro-instrument for coordinated sensing. *Sensors* **2002**, *2*, 270–285.
12. Cox, S. *Observations and Measurements—Part 1—Observation Schema*. Implementation Standard 07-022r1, Open Geospatial Consortium Inc., 2007. Available online: <http://www.opengeospatial.org/standards/om> (accessed on 31 August 2011).
13. Botts, M.; Robin, A. *OpenGIS® Sensor Model Language (SensorML) Implementation Specification*; Implementation Specification 1.0.0, Open Geospatial Consortium Inc., 2007. Available online: http://portal.opengeospatial.org/files/?artifact_id=21273 (accessed on 31 August 2011).
14. Harris, S. *OpenGIS Transducer Markup Language (TML) Implementation Specification*; Implementation Specification 06-010r6, Open Geospatial Consortium Inc., 2007. Available online: <http://www.opengeospatial.org/standards/tml> (accessed on 31 August 2011).
15. Na, A.; Priest, M. *Sensor Observation Service*; Implementation Standard OGC 06-009r6, Open Geospatial Consortium Inc., 2007. Available online: <http://www.opengeospatial.org/standards/sos> (accessed on 31 August 2011).
16. Simonis, I. *OpenGIS® Sensor Planning Service Implementation Specification*; Implementation Specification OGC 07-014r3, Open Geospatial Consortium Inc., 2007. Available online: <http://www.opengeospatial.org/standards/sps> (accessed on 31 August 2011).
17. Simonis, I. *OGC Sensor Alert Service Candidate Implementation Specification*; Implementation Specification 06-028r3, Open Geospatial Consortium Inc., 2006. Available online: http://portal.opengeospatial.org/files/?artifact_id=15588 (accessed on 31 August 2011).
18. Simonis, I.; Wytzisk, A. *Web Notification Service*; Discussion Paper 03-008r2, Open Geospatial Consortium Inc., 2003. Available online: http://portal.opengeospatial.org/files/?artifact_id=1367 (accessed on 31 August 2011).
19. Schimak, G.; Havlik, D. Sensors anywhere—sensor web enablement in risk management applications. *ERCIM News* **2009**, *76*, 40–41.
20. Bleier, T.; Božić, B.; Bumerl-Lexa, R.; da Costa, A.; Costes, S.; Iosifescu, I.; Martin, O.; Frysinger, S.; Havlik, D.; Hilbring, D.; Jacques, P.; Klopfer, M.; Kunz, S.; Kutschera, P.; Lidstone, M.; Middleton, S.E.; Roberts, Z.; Sabeur, Z.; Schabauer, J.; Schlobinski, S.; Shu, T.; Simonis, I.; Stevenot, B.; Usländer, T.; Watson, K.; Wittamore, K. *SANY An Open Service Architecture for Sensor Networks*; SANY Consortium, 2009. Available online: <http://www.sany-ip.eu/publications/3317> (accessed on 31 August 2011).
21. Fielding, R.T. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, Information and Computer Science, University of California, Irvine, CA, USA, 2000.
22. Pautasso, C.; Zimmermann, O.; Leymann, F. Restful Web Services vs. “Big” Web Services: Making the Right Architectural Decision. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008*, Beijing, China, 21–25 April 2008; pp. 805–814.
23. Pautasso, C.; Wilde, E. Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, Madrid, Spain, 20–24 April 2009; pp. 911–920.

24. Havlik, D.; Bleier, T.; Schimak, G. Sharing senso data with sensorsa and cascading sensor observation service. *Sensors* **2009**, *9*, 5493–5502.
25. Nebert, D.; Whiteside, A.; Vretanos, P.A. OpenGIS Catalogue Services Specification. Implementation Specification 07-006r1, Open Geospatial Consortium Inc., 2007. Available online: http://portal.opengeospatial.org/files/?artifact_id=20555 (accessed on 31 August 2011).
26. Jirka, S.; Bröring, A.; Stasch, C. Discovery mechanisms for the sensor web. *Sensors* **2009**, *9*, 2661–2681.
27. Jirka, S.; Nust, D.; Schulte, J.; Houbie, F. Integrating the OGC Sensor Web Enablement Framework into the OGC Catalogue. In *Proceedings of the 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services*, Como, Italy; 26–27 August 2010, 2010.
28. Galpin, I.; Brenninkmeijer, C.Y.A.; Gray, A.J.G.; Jabeen, F.; Fernandes, A.A.A.; Paton, N.W. SNEE: A query processor for wireless sensor networks. *Distrib. Parallel Databases* **2011**, *29*, 31–85.
29. Brenninkmeijer, C.Y.A.; Galpin, I.; Fernandes, A.A.A.; Paton, N.W. A Semantics for a Query Language over Sensors, Streams and Relations. In *Proceedings of 25th British National Conference on Databases, BNCOD 25*, Cardiff, UK, 7–10 July 2008; Volume 5071, pp. 87–99.
30. Calbimonte, J.P.; Corcho, Ó.; Gray, A.J.G. Enabling Ontology-Based Access to Streaming Data Sources. In *Proceedings of the 9th International Semantic Web Conference, ISWC 2010*, Shanghai, China, 7–11 November 2010; Volume 6496, pp. 96–111.
31. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New generation sensor web enablement. *Sensors* **2011**, *11*, 2652–2699.
32. Sheth, A.P.; Henson, C.; Sahoo, S.S. Semantic sensor web. *IEEE Int. Comput.* **2008**, *12*, 78–83.
33. Bröring, A.; Maué, P.; Janowicz, K.; Nüst, D.; Malewski, C. Semantically-enabled sensor plug & play for the sensor web. *Sensors* **2011**, *11*, 7568–7605.
34. Corcho, O.; Garcia-Castro, R. Five challenges for the semantic sensor web. *Semant. Web J.* **2010**, *1*, 121–125.
35. Page, K.R.; De Roure, D.C.; Martinez, K.; Sadler, J.D.; Kit, O.Y. Linked Sensor Data: RESTfully Serving RDF and GML. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks*, Washington, DC, USA, 25–29 October 2009; pp. 49–63.
36. Patni, H.; Henson, C.; Sheth, A.P. Linked Sensor Data. In *Proceedings of the International Symposium on Collaborative Technologies and Systems, CTS 2010*, Chicago, IL, USA, 17–21 May 2010; pp. 362–370.
37. Power, D.; Burstein, F.; Sharda, R. Reflections on the Past and Future of Decision Support Systems: Perspective of Eleven Pioneers. In *Decision Support: An Examination of the DSS Discipline*; Springer: Berlin, Heidelberg, Germany, 2011; Volume 14, pp. 25–48.
38. Drummond, J.; João, E.; Billen, R. Current and Future Trends in Dynamic and Mobile GIS. In *Dynamic and Mobile GIS*; CRC Press: Boca Raton, FL, USA, 2007; pp. 289–300.
39. Shim, J.; Warkentin, M.; Courtney, J.; Power, D.; Sharda, R.; Carlsson, C. Past, present, and future of decision support technology. *Decis. Support Syst.* **2002**, *33*, 111–126.
40. FLOOD WATCH—Decision Support System for Real-Time Forecasting. Available online: <http://www.dhigroup.com/SolutionSoftware/FLOODWATCH.aspx> (accessed on 18 August 2011).

41. Coastal Flooding. Available online: <http://mikebydhi.com/Applications/CoastAndSea/CoastalFlooding.aspx> (accessed on 18 August 2011).
42. Babitski, G.; Bergweiler, S.; Grebner, O.; Oberle, D.; Paulheim, H.; Probst, F. SoKNOS—Using Semantic Technologies in Disaster Management Software. In *ESWC 2011, Part II*; Springer: Heraklion, Crete, Greece, 2011; Volume 6644, pp. 183–197.
43. Casola, V.; D’Onofrio, L.; Lorenzo, G.D.; Mazzocca, N. A Service-Based Architecture for the Interoperability of Heterogeneous Sensor data: A Case Study on Early Warning. In *Geographic Information and Cartography for Risk and Crisis Management*, 1st ed.; Lecture Notes in Geoinformation and Cartography; Springer: Berlin-Heidelberg, Germany, 2010; pp. 249–263.
44. Jirka, S.; Broering, A.H.; Stasch, C. Applying OGC Sensor Web Enablement to Risk Monitoring and Disaster Management. In *Proceedings of the Workshop on Sensorweb Enablement: Strengthening the SDI at the GSDI 11 World Conference*, Rotterdam, The Netherlands, 15–19 June 2009.
45. Channel Coastal Observatory. Available online: <http://www.channelcoast.org/> (accessed on 18 August 2011).
46. SemSorGrid4Env Ontology Network for Environment Decision Support. Available online: <http://www.sensorgrid4env.eu/ontologies/> (accessed on 31 August 2011).
47. Lefort, L.; Henson, C.; Taylor, K. *Semantic Sensor Network XG Final Report*; Incubator Group Final Report, W3C, 2011. Available online: <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/> (accessed on 31 August 2011).
48. W3C Semantic Sensor Network Incubator Group. Available online: <http://www.w3.org/2005/Incubator/ssn/> (accessed on 18 August 2011).
49. DOLCE+DnS UltraLite Ontology. Available online: <http://www.loa-cnr.it/ontologies/DUL.owl> (accessed on 18 August 2011).
50. Pérez de Laborda, C.; Conrad, S. Relational.OWL: A Data and Schema Representation Format Based on OWL. In *Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling, APCCM 2005*, Newcastle, Australia, January 2005; pp. 89–96.
51. Raskin, R.G.; Pan, M.J. Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). *Comput. Geosci.* **2005**, *31*, 1119–1125.
52. Geographic information—Services. International Standard ISO19119:2005, ISO, 2005.
53. Martin, D.; Burstein, M.; Hobbs, J.; Lassila, O.; McDermott, D.; McIlraith, S.; Narayanan, S.; Paolucci, M.; Parsia, B.; Payne, T.; Sirin, E.; Srinivasan, N.; Sycara, K. OWL-S: Semantic Markup for Web Services. Member Submission, W3C, 2004. Available online: <http://www.w3.org/Submission/OWL-S> (accessed on 31 August 2011).
54. Roman, D.; Keller, U.; Lausen, H.; de Bruijn, J.; Lara, R.; Stollberg, M.; Polleres, A.; Feier, C.; Bussler, C.; Fensel, D. Web service modeling ontology. *Appl. Ontol.* **2005**, *1*, 77–106.
55. Ordnance Survey Ontologies. Available online: <http://www.ordnancesurvey.co.uk/oswebsite/ontology/> (accessed on 18 August 2011).
56. Christensen, E.; Curbera, F.; Meredith, G.; Weerawarana, S. *Web Services Description Language (WSDL) 1.1*; Note, W3C, 2001. Available online: <http://www.w3.org/TR/wsdl> (accessed on 31 August 2011).

57. Gudgin, M.; Hadley, M.; Mendelsohn, N.; Moreau, J.J.; Nielsen, H.F.; Karmarkar, A.; Lafon, Y. *SOAP Version 1.2 Part 1: Messaging Framework (2nd ed)*; Recommendation 1.2, W3C, 2007. Available online: <http://www.w3.org/TR/soap12-part1/> (accessed on 31 August 2011).
58. Associated British Ports. Available online: <http://www.abports.co.uk/> (accessed on 18 August 2011).
59. Gray, A.J.G.; Galpin, I.; Fernandes, A.A.A.; Paton, N.W.; Page, K.; Sadler, J.; Kyzirakos, K.; Koubarakis, M.; Calbimonte, J.P.; García-Castro, R.; Corcho, O.; Gabaldón, J.E.; Aparicio, J.J. SemSorGrid4Env Architecture—Phase II. Deliverable D1.3v2, SemSorGrid4Env, 2010.
60. Koubarakis, M.; Kyzirakos, K. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In *Proceedings of the 7th Extended Semantic Web Conference, ESWC 2010*, Heraklion, Crete, Greece, 30 May–3 June 2010; pp. 425–439.
61. Open Geospatial Consortium Incorporated. *OpenGIS Implementation Standard for Geographic information—Simple feature access—Part 1: Common Architecture*; Implementation Standard 1.2.1, Open Geospatial Consortium Inc., 2010. Available online: http://portal.opengeospatial.org/files/?artifact_id=25355 (accessed on 31 August 2011).
62. Perry, M.; Herring, J. *GeoSPARQL—A Geographic Query Language for RDF Data*; Proposal for an OGC Draft Candidate Standard 11-052r1, Open Geospatial Consortium Inc., 2011. Access restricted to OGC members.
63. Perry, M. A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data. PhD thesis, College of Engineering and Computer Science, Wright State University, Dayton, Ohio, USA, 2008.
64. Kyzirakos, K.; Karpathiotakis, M.; Koubarakis, M. Developing Registries for the Semantic Sensor Web using stRDF and stSPARQL (Short Paper). In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks, SSN 2010*, Shanghai, China, 7–11 November 2010.
65. LinkedGeoData. Available online: <http://linkedgeodata.org/> (accessed on 25 August 2011).
66. GeoSPARQL SWG. Available online: <http://www.opengeospatial.org/projects/groups/geosparqlswg> (accessed on 18 August 2011).
67. Kolas, D.; Self, T. Spatially Augmented Knowledgebase. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, ISWC2007+ASWC2007*, Busan, Korea, 11–15 November 2007; Volume 4825, pp. 785–794.
68. Kolas, D. Supporting Spatial Semantics with SPARQL. In *Proceedings of the Terra Cognita Workshop, Terra Cognita 2008*, Karlsruhe, Germany, 26 October 2008.
69. Batsakis, S.; Petrakis, E.G.M. SOWL: Spatio-Temporal Representation, Reasoning and Querying over the Semantic Web. In *Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS 2010*, Graz, Austria, 1–3 September 2010.
70. Brodt, A.; Nicklas, D.; Mitschang, B. Deep Integration of Spatial Query Processing into Native RDF Triple Stores. In *Proceedings of the 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2010*, San Jose, CA, USA, 3–5 November, 2010; pp. 33–42.

71. de la Beaujardiere, J. *OpenGIS® Web Map Server Implementation Specification*; Standard Specification 06-042, Open Geospatial Consortium Inc., 2006. Available online: http://portal.opengeospatial.org/files/?artifact_id=14416 (accessed on 31 August 2011).
72. Antonioletti, M.; Atkinson, M.; Krause, A.; Laws, S.; Malaika, S.; Paton, N.W.; Pearson, D.; Riccardi, G. *Web Services Data Access and Integration—The Core (WS-DAI) Specification*; Version 1.0. Recommendation GFD.74, Open Grid Forum, 2006. Available online: <http://www.ogf.org/documents/GFD.74.pdf> (accessed on 31 August 2011).
73. WS-DAIR 1.0. Available online: <http://sourceforge.net/projects/ogsa-dai/files/OGSA-DAI%20WS-DAIR%201.0/> (accessed on 18 August 2011).
74. Vretanos, P.A. *Web Feature Service Implementation Specification*; Standard Specification 04-094, Open Geospatial Consortium Inc., 2004.
75. Berners-Lee, T. *Linked Data*; Technical Report; W3C, 2006. Available online: <http://www.w3.org/DesignIssues/LinkedData.html> (accessed on 31 August 2011).
76. Apache CXF JavaScript Clients. Available online: <https://cwiki.apache.org/CXF20DOC/javascript-clients.html> (accessed on 18 August 2011).
77. Page, K.R.; De Roure, D.C.; Martinez, K. REST and Linked Data: A Match Made for Domain Driven Development? In *Proceedings of the 2nd International Workshop on RESTful Design, WS-REST-2011*, Hyderabad, India, 28 March 2011; pp. 22–25.
78. Page, K.R.; Frazer, A.J.; Nagel, B.J.; De Roure, D.C.; Martinez, K. Semantic Access to Sensor Observations through Web APIs. In *Proceedings of the 5th IEEE International Conference on Semantic Computing, ICSC-2011*, Palo Alto, CA, USA, 18–21 September 2011.
79. Janowicz, K.; Compton, M. The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In *Proceedings of the 3rd International workshop on Semantic Sensor Networks, SSN10*, Shanghai, China, 7–11 November 2010.
80. Barnaghi, P.; Presser, M.; Moessner, K. Publishing Linked Sensor Data. In *Proceedings of the 3rd International Workshop on Semantic Sensor Networks, SSN10*, Shanghai, China, 7–11 November 2010.
81. Pschorr, J.; Henson, C.; Patni, H.; Sheth, A. Sensor Discovery on Linked Data. In *Proceedings of the 7th Extended Semantic Web Conference, ESWC2010*, Heraklion, Greece, 30 May–2 June 2010.
82. Le-Phuoc, D.; Hauswirth, M. Linked Open Data in Sensor Data Mashups. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks 2009, SSN09*, Washington, DC, USA, 26 October 2009.
83. Janowicz, K.; Bröring, A.; Stasch, C.; Everding, T. Towards Meaningful URIs for Linked Sensor Data. In *Proceedings of the Workshop at Future Internet Symposium 2010: Towards Digital Earth: Search, Discover and Share Geospatial Data, FIS-DE 2010*, Berlin, Germany, 20 September 2010.
84. Pullen, T.; Allsop, N.W.H.; Bruce, T.; Kortenhaus, A.; Shüttrumpf, H.; van der Meer, J.W. *EurOtop: Wave Overtopping of Sea Defences and Related Structures: Assessment Manual*; Vlaams Instituut voor de Zee InnovOcean site Wandelaarkaai 7 B-8400 OOSTENDE, Belgie. 2007.

85. Bates, P.D.; Dawson, R.J.; Hall, J.W.; Horritt, M.; Nicholls, R.J.; Wicks, J.; Hassan, M.A.A.M. Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coast. Eng.* **2005**, *52*, 793–810.
86. Ext JS 4. Available online: <http://www.sencha.com/products/extjs/> (accessed on 18 August 2011).
87. GeoExt. Available online: <http://www.geoext.org/> (accessed on 18 August 2011).
88. Open Layers. Available online: <http://openlayers.org/> (accessed on 18 August 2011).
89. Automatic Identification System Overview. Available online: <http://www.navcen.uscg.gov/?pageName=AIS> (accessed on 24 June 2011).

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).