Simulation Modelling Practice and Theory 19 (2011) 1745–1757

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A semantic enriched data model for sensor network interoperability

Flora Amato, Valentina Casola, Andrea Gaglione, Antonino Mazzeo

Dipartimento di Informatica e Sistemistica, Universita' degli Studi di Napoli "Federico II", Naples, Italy

ARTICLE INFO

Article history:
Available online 8 October 2010

Keywords:
Sensor network interoperability
Sensor data model
Service based architectures
Web services

ABSTRACT

One of the main open issues in the development of applications for sensor network management is the definition of interoperability mechanisms among the several monitoring systems and heterogeneous data. Interesting researches related to integration techniques have taken place, they are primary based on the adoption of sharing data mechanisms. In the last years, the Service-Oriented Architecture (SOA) approach has become predominant in many sensor network projects as it enables the cooperation and interoperability of different sensor platforms at an higher level of abstraction. In this paper we propose a novel architecture for the interoperability of sensor networks, which is based on web services technologies and on a common data model enriched with semantic concepts and annotations. The proposed architecture allows the development of complex decision support system applications by integration of heterogeneous data, accessible through services, according to standard data format and standard protocols.

1. Introduction

The large diffusion of sensor systems, together with their numerous applications has led to a huge heterogeneity in the logic for interfacing and collecting data from these systems.

One of the main open issues in the environmental monitoring and management is the integration of heterogeneous data from different sources to manage and elaborate risk mitigation strategies, including emergency management planning.

However in order to automatize the elaboration of such data, specific integration frameworks for accessing different data sources are needed. These frameworks should be able to access data, sensed by different sensing infrastructures, they should hide the heterogeneity of different sensor systems (in terms of sensors, networks or middleware technologies) and provide a standard way to access them. Interesting researches related to integration techniques for heterogeneous sensor networks have taken place, but nowadays only few architectures have been proposed. Most of them try to define a common exchange mechanism among different sensor systems in order to facilitate the integration and provide a software integration layer which allows different sensor systems to collaborate for the same purpose. Very often these solutions are tightly related to proprietary technologies and they are neither scalable nor open.

In the last years, the Service-Oriented Architecture (SOA) approach has become a cornerstone in many sensor networks projects. SOA-approach enables the cooperation and interoperability of different sensor platforms as it provides discovery, access and sharing of the services, data, computational and communication resources by the adoption of open standards. New open standards enable the definition of cooperative environments, they are based on the adoption of common data models to formally define and represent data knowledge. As an example of data model, the OpenGeospatial Consortium provides an XML schema (the Sensor Model Language) for defining geometric, dynamic and observational characteristics of sensors and other standards as Observation and Measurement to describe observed phenomenon [1]. The standards:

E-mail addresses: flora.amato@unina.it (F. Amato), casolav@unina.it (V. Casola), andrea.gaglione@unina.it (A. Gaglione), mazzeo@unina.it (A. Mazzeo).

1569-190X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2010.09.010

(i) provide general sensor information in order to support of data discovery, (ii) support the processing and analysis of sensor measurements, (iii) support the geolocation of the measured data, (iv) provide performance characteristics (e.g. accuracy, threshold, etc.), and (v) archive fundamental properties and assumptions regarding sensors. Indeed, these standards focus on syntactic aspects of interoperability issues.

In this paper, a service based architectural model to provide interoperability among different sensor networks is proposed and a semantic based data model for sensors is introduced and discussed in details. As already said, the strong advantage of web service architectures is to build information systems enabling the creation of applications by combining loosely coupled and interoperable services without the knowledge of the underlying systems. On the other hand, the semantic enriched data model grants interoperability among multi-technology systems, providing a formal model for the integration of data gathered by different and heterogeneous sources. A formal data model specifies data relations, terminology and meanings, it is implemented through a set of ontologies formally described in Web Ontology Language (OWL). In this work, we will present an innovative architecture for risk management that is completely based on services; it allows the integration of heterogeneous data and the implementation of applicative standard Web Services to let data (raw or aggregated) be available to authorized end-users and/or applications that need them. Then we will describe a data collection service with the definition of a common data model for specific sensors. The analyzed data are related to seismic, volcanic, pluviometric and meteo-marine sensors, they will be formalized according to the Observation and Measurement standard [20] and will be enriched with semantic information.

We will illustrate how the formalized data can be accessible through a standard web services and can be invocated by other services for integration and elaboration purposes. The reminder of the paper is structured as follows: in Section 2 some related works are reported to assess the state of the art of many methodologies and technologies to face interoperability among heterogeneous data. In Section 3 we will present the architectural model by illustrating the main layers of the proposed architecture and their functionalities. In Sections 4 and 5 we will illustrate the proposed data model that has been enriched with a semantic approach and we will discuss in details the specific services that are needed to collect and integrate data from different sensor networks. Finally in Section 6 some concluding remarks will be given.

2. Related works

The need to guarantee interoperability among several monitoring systems and integration of heterogeneous data, can be seen from different point of views: (i) *Syntactic*, to overcome technical heterogeneity; (ii) *Semantic*, to overcome ambiguities and different interpretations; (iii) *Application*, to deliver sustainable and re-usable concepts and components of the application domain; and (iv) *Phenomena observation*, to evaluate the meaning of the observation from temporal, spatial and thematic perspectives. Some solutions to specific, but not complete, aspects of interoperability are available in the literature, the OGC proposed Sensor Web Enablement (SWE) [1], a suite of specifications to model sensor characteristics and services. In particular, the suite includes: (i) Sensor Model Language (Sensor ML), (ii) Observation & Measurement and (iii) Sensor Observation Service. They allow to model sensor and sensor observations, data retrieval mechanism and web services (for access of the sensor data via web); it is possible to specify information as coordinates and timestamps, but they do not allow to state the intrinsic relations among data and the meaning of sensor observations, making difficult the interoperability, the evaluation of the phenomena and the detection of situation awareness [7]. A first attempt of semantics definition for sensor Web has been proposed in the SSW framework [8], in which enhanced meaning for sensor observations is given by adding semantic annotations to existing standard sensor languages of the SWE, in order to increase interoperability and provide contextual information for situation awareness. Several efforts have been done in the data modeling field too: different ontologies for heterogeneous sensor data representation are proposed. The ontology presented by Ceruti in [14] models different concepts, as platforms and sensors, as characteristics tangible and intangible, and relations and concepts such as data combinations. Another ontology for the sensor networks, presented by Eid et al. in [16], provides semantic representation of sensor networks data, aiming at interpreting unambiguous structured information.

From the architectural interoperability point of view the current leading approach is based on Service Oriented Architectures (SOA). In [17] the authors introduced Web Service Resource Framework (WSRF) mechanisms within the core services implementation of the NICTA Open Sensor Web Architecture (NOSA). WSRF enables to perform simultaneous observational queries to heterogeneous Sensor Networks. Moreover, the GeoICT group at York University have built an OGC SWE compliant Sensor Web infrastructure [18], developing a Sensor Web client to visualize geospatial data [11], and a set of stateless Web Services called GeoSWIFT [19]. Furthermore, two main European early warning projects based on services have been proposed; the Wide Information Network (WIN) [10,9] and ORCHESTRA [9] projects. WIN aims at developing an open and flexible platform to support multi-hazard and risk domains at European level, integrating national and regional data flows in the frame of a Web Service Architecture, and proposing a set of generic services and standard data modeling components that can be used in the deployment of several cases. The WIN Metadata Model is based on existing standards, such as Dublin Core [3], GML [4] and ISO19115 [5], and provides additional specifications for WIN. It allows to manage in uniform way all metadata by using the standard Electronic Business Registry Information Model (ebRIM). The WIN Data models are implemented via the Data Web Service component and the metadata via a ebXML Registry tool. Both components and tools use the standard SOAP/WSDL type of Web Services. An the other hand, ORCHESTRA adapts the ISO/IEC 10746 Reference Model for Open Distributed Processing to service-oriented architectures [6]; adapting, in particular, the Reference Model for geospatial service networks on a process model compliant with the ISO standard RM-ODP, in order to design and implement geospatial SOAs. The ORCHESTRA architecture uses W3C Web services platform and the Geography Mark-up Language (GML) to implement web services.

Different methods and techniques have been developed in order to assure the interoperability among several monitoring systems and integration of heterogeneous data. They represent solutions to specific interoperability and/or integration problems, but, at best of our knowledge, a comprehensive proposal of an architecture that provide a solution for the complete gamma of needs (i.e. able to manage phenomena observation, interoperability among several monitoring systems and integration of heterogeneous data, and that can be easily integrated with other pre-existing platform) does not exist, yet.

3. A service-based architecture for sensor network integration

To design an open system and manage heterogeneous sensor data sources we propose a Service Oriented Architecture based on Web Services technologies. The adoption of standard technologies such as HTTP, XML, SOAP, WSDL, and UDDI [2] enables pervasive adoption and deployment of web services to reach interoperability. In fact, Web services are Internet based applications that communicate with other applications to offer data or services in a standard way.

The proposed architecture offers several services and functionalities, which can be classified into two main categories: (a) services to elaborate data from heterogeneous sources; (b) services to manage all data sources (sensors, databases, simulators, etc.). We have located three horizontal layers: Data Collection Service Layer, Integration Service Layer, Application Service Layer as illustrated in Fig. 1, furthermore some transversal layers can be introduced to provide Security, Management and Interoperability services. Each layer offers different services and is characterized by a deep functional specialization; furthermore, each layer is able to communicate with other layers through standard WSDL interfaces.

Grouping similar services in different layers offers different advantages: most complex services are built upon the collaboration of more elements of smaller complexities; from a security point of view each service can have a specific security policy to control access and preserve data privacy. Such approach enables the building of a more robust, scalable and maintainable architecture, with a better logical structure, too.

In the following, we provide a brief description of every layer and their specific services:

1. *Data Collection Service Layer:* sensor networks have specific and different querying languages and paradigms, hence the Data Collection Service Layer provides to the upper layers a homogeneous view of the networks. At this aim, a data model of sensors, measurements and phenomena is required in order to properly share the relevant information of observed phenomena. The innovative approach that we propose aims at enriching sensed data with ontological modeling techniques, mapping sensor data sets from a raw representation given by the sensor hardware/middleware to an XML/RDF data model and to an O&M standard representation. Indeed, the services offered in this layer translate and semantically enrich proprietary data format in a common data model as illustrated in next sections.

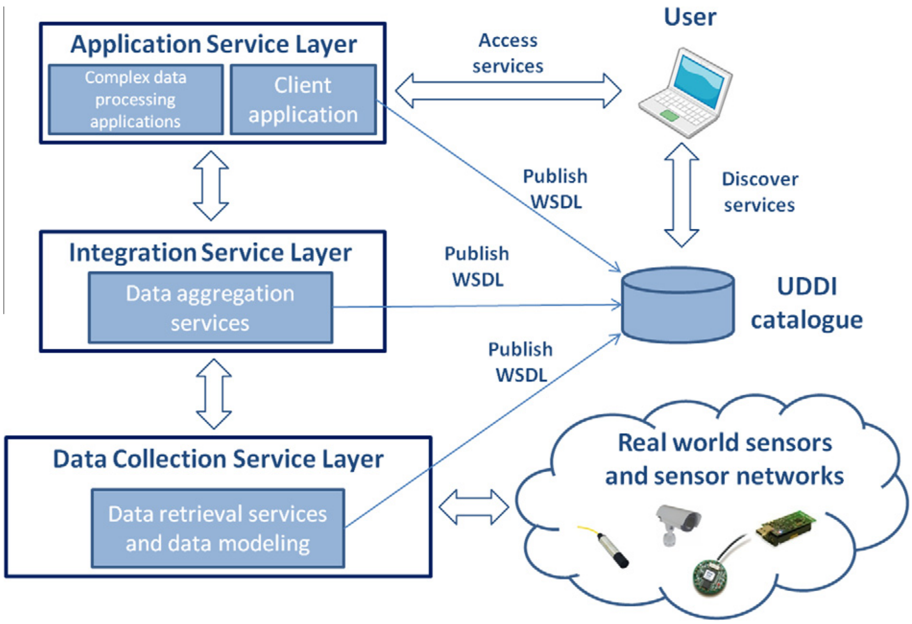

Fig. 1. An overview of the architecture.

2. *Integration Service Layer:* the services of this layer provide aggregated data sets to the application level; in particular, it integrates different networks that are observing the same phenomena and must be aggregated to better understand complex phenomena. The services clusterize network data sets according to the defined data model. Let us consider for example a typical scenario where different technologies are adopted to sense the same data; while the collection layer is responsible to retrieve such data and represent them, this layer is capable of integrating data related to the same phenomena but observed by different networks.
3. *Application Service Layer:* different kinds of applications can be implemented for monitoring and elaborating sensed data. They can implement decision support systems as warning threshold models or real-time event notification. Applications are built by invoking and composing services defined in other layers; they all elaborate complex data structures formatted according to the defined data model and accessible via standard WSDLs.

Each service has its own security policy and it is published in a public registry.

We note that sensed data, which are stored and used to protect critical infrastructures, may represent sensible information, this implies that such data have to be managed only by authorized entities. At this aim, the security services (not reported in the figure) enforce WS-Security standards to grant access (authentication and authorization) only to those authorized user. These services also deal with confidentiality and data integrity of the messages exchanged, non-repudiation of requests of messages and resilience to denial-of-service attacks.

In the following we will focus our attention on the Integration and Data Collection Service Layers. We will illustrate a service-oriented architecture able to support functionalities of both layers and propose a common data model enhanced with semantic annotations for the representation of both sensor data and sensor systems.

4. Semantic based data model

Proper data modeling is an essential task for providing an efficient and effective representation of the phenomena of interest. Furthermore it is a crucial step for achieving full interoperability among heterogeneous sensor networks.

Defining and adopting models is necessary in order to catch the information of interest together with the formats and the properties that have to be represented. Hence, for our aims, one of the main purposes of the modeling process is the collection and structuring of information gathered by sensors that can be in a raw format.

The basic problem to be addressed, when integrating data from heterogeneous sources, come from syntactic, schematic and semantic differences in the representation. *Syntactic* heterogeneity refers to differences among paradigms used to format data such as plain text, relational DB or xml documents. *Schematic* heterogeneity refers to different aggregation or generalization hierarchy defined for the same phenomena. Finally, *Semantic* heterogeneity regards disagreement on the meaning or interpretation on same data [12,13].

Our proposal aims at facing and solving these differences. In particular, syntactic heterogeneity is resolved by converting proprietary data formats into an XML [25] document; schematic heterogeneity is faced by adopting the standard O& M Language [20]; finally semantic heterogeneity is faced by defining a common data model formally described using an ontology.

The adoption of an ontology-based formalism is motivated by the need of giving a shared, semantic based, model of the information to be elaborated. In addition, many recent works have shown that precision in service discovery increases when semantic representation is used instead of syntactic one [15]. We have then defined an ontological common data model, to describe sensor data and observations in a unique and unambiguous way.

In the Sensor Domain ontology we recur to data semantic modeling in order to formalize an explicit representation of data characteristics and to locate relations and similitude among data sensed by heterogeneous sources that cannot be directly derived by the syntactical structure of the data. The proposed model is coded in RDF triples in compliance with the Semantic Sensor Web [21,22] approach, too.

The adoption of an ontology helps to fuse data and meaning, so, for the model description, we adopted open standards for interoperability, enriched with semantic information. In fact, a semantic based technology allows us to deal with raw

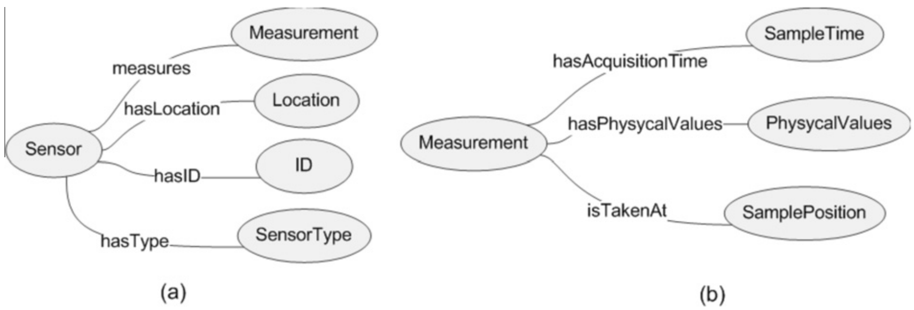

Fig. 2. The sensor and measurement classes.

data and manage them as a global Knowledge Base, we obtained an explicit representation of the meaning of data and services that is useful for extracting relevant information and for integrating them. The knowledge can be managed as a database that can be queried in a structured way enabling advanced operation as logic reasoning. Furthermore, it is possible to enforce consistency verification on modeled data to verify the compliance to the model and the acceptability of sensed values.

The resulting ontological model is made of two main classes: the **Sensor Class** includes knowledge about sensor specifications; the **Measurement Class** defines information about the data collected by the different sensors.

As illustrated in Fig. 2a, the Sensor Class models the sensor characteristic such as Location, Identification, Measurement, Sensor Type. Among the others, the Location entity describes geographic information by using coordinates in DMS or UTM system while the Sensor Type entity specifies information about sensor typology description.

The Measurement Class, as illustrated in Fig. 2b, contains information about physical characteristics of gathered samples. Every measurement is characterized by a set of physical parameters as the SampleTime, that indicates the sample time period of phenomena observations, and the SamplePosition that indicates the position of the samples at the beginning of the observation, specified in the same data-format of the Location. In addition, PhysicalValues are expressed in terms of statistic information (as mean values and covariance) and are aided with sensor parameters (as sensor power, voltage, bearing and direction) that provide the accuracy and the measurement units, defined in external standard models.[1]

Note that such semantic information cannot be derived by analyzing the sensor data or specified using the SWE languages; those standards, in fact, allow to characterize physic concepts, such as spatial and temporal coordinates, but not to specify the meaning of sensor observations, that are necessary, in order to allow the interoperability of heterogeneous sensors and improve situation awareness.

The proposed model can be extended with external ontologies to add concepts and relationships and to give an enriched description of sensors, observations and phenomena; for example, it can be extended to model how sensors are grouped into classes according to dynamic associations defined by specific phenomena under observation. Let us consider for instance three different sensor types (temperature, humidity and wind sensor) that may be placed in the same area and can be dynamically grouped to monitor the same phenomena in a collaborative way for weather forecasting purposes.

Finally, thanks to this model, it is possible to share and standardize, at a higher level in the service, the data format and so resolve possible conflicts generated by heterogeneous sources.

5. Integration service and data collection service layers

In this section we will illustrate a service oriented architecture, named SeNsIM-Web [23], covering the functionalities of both the Integration Service and the Data Collection Service Layers. In order to cope with heterogeneity of sensor data, our framework is based on:

- an *architectural model* able to support in an efficient way the management of such data even when sensed by different networks;
- a *data model* able to represent sensor data in a unique logical view thanks to a semantic enrichment, thus allowing a simple management within the framework.

The architectural model has been designed by exploiting the wrapper–mediator paradigm, a well-known technique to integrate data from heterogeneous source [26]; according to this model a mediator can access different data sources by means of ad hoc connectors (wrapper components). In a typical working scenario, each wrapper explores and monitors the local sensor network and sends to the mediator an appropriate description of the related information according to a common data model. On the other hand the mediator integrates and organizes such information by keeping a unique view of all systems in order to satisfy user or application queries. The data model aims to represent both sensor nodes and sensor networks by combining structural and behavioral description of sensors. The original data model of SeNsIM-Web has also been enhanced with semantic annotations, as described in the next section. In order to have an open and standard architecture we provide both wrapper and mediator components with web service interfaces in such a way that the mediator implements functionalities of the Integration Service Layer, while a generic wrapper implements those of the Data Collection Service Layer. The reference architecture is shown in Fig. 3.

The mediator provides the Service Integration layer and its services are invoked by a generic user or an application according to its WSDL interface and SOAP protocol. The mediator component aims to classify networks features as well as format and forward queries to specific wrappers; a DBMS is used to store data related to networks with their sensors, queries and related results. SOAP is also used for any communication between the mediator and the wrappers according to their provided services.

Fig. 4 shows all services provided by the mediator and wrappers as well as their interactions. Below, we will report a brief description of the mediator services and in Figs. 5–7 there are some illustrative UML sequence diagrams to show the interaction with both the client and the wrapper services:

[1] As the ontology model defined by NASA for measurement description at http://sweet.jpl.nasa.gov/2.0/math.owl.

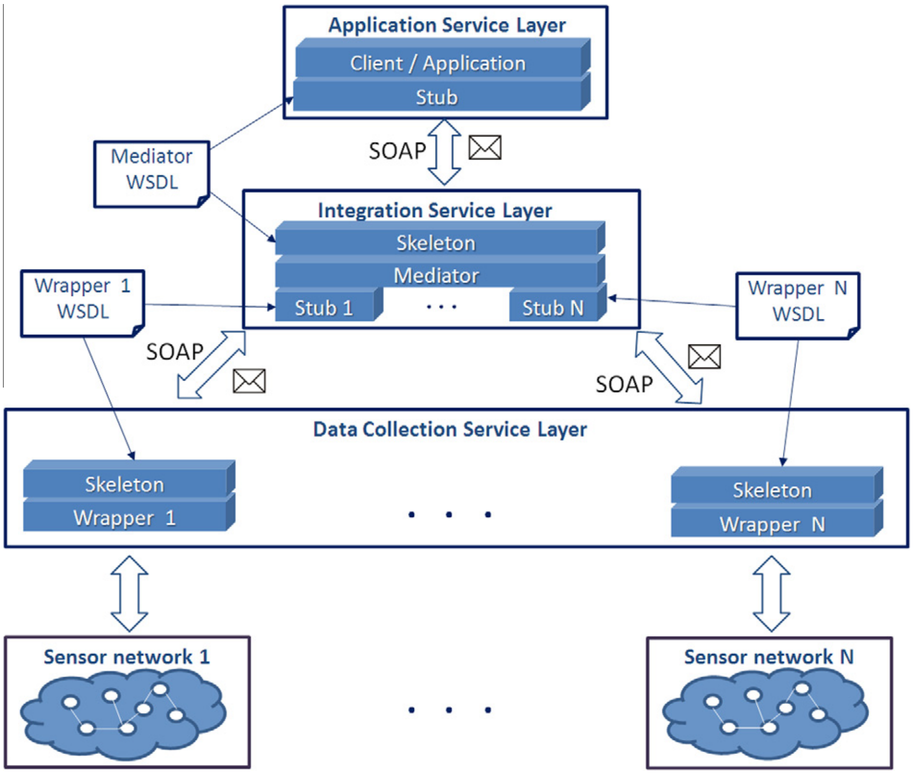

Fig. 3. SeNsIM-web collection and integration layers.

1. *getConnection* allows for the discovery and registration of a specific wrapper. At this aim, it invokes the *activation* service provided by the wrapper (see the sequence diagram in Fig. 5).
2. *getNetworks* gives as output the list of all networks connected to the system, their topology and basic information such as network identifiers, number of sensors and maximum depth for each network.
3. *getNetworkFeatures* gets as input parameter the network identifier and gives in output: system description, type of middleware, number of sensors, maximum depth. Further, it gives information about the wrapper which is handling the network (IP address, registration time and communication ports). Finally it gives information about the base station, its working frequency and its communication link to the wrapper (Fig. 6).
4. *getParameters* gets as input a specific sensor identification given by the triple: network identifier, cluster (if defined) identifier and sensor identifier. It gives as output the intrinsic parameters of the sensor, such as the available memory, the voltage and the channel quality.
5. *getPredicates* gets as input a specific sensor identification and gives as output the sensor predicates, i.e. the physical variable measured by the sensor.
6. *Monitoring* carries out a simple monitoring task by querying a sensor or a whole network. It gets as input the query parameters: sensor identification, sample period, duration, the retrieval interval, already defined in SeNsIM [24] as the time interval in which wrappers may collect query results before sending them to the mediator. As shown in Fig. 7, the query process can start by invoking the *requestProcessing* service provided by the wrapper. The results are sent back to the mediator in a SOAP response message via the *getResults* wrapper service which is invoked again by the Monitoring service. Finally data are stored in the mediator database and shown to the user.

On the wrapper side, the services provided are:

1. *activation*, invoked by the *getConnection* service, allows for the registration of a specific wrapper within the system. In particular, the wrapper injects a *discovery query* to the underlying network and subsequently builds a description of the network which is sent back to the mediator (Fig. 5).
2. *requestProcessing*, invoked by the *Monitoring* service, aims to submit queries to the underlying network.
3. *getResults*, invoked by the *Monitoring* service, allows getting results related to a specific query (Fig. 7).

As already said, each wrapper of the Data Collection Service Layer aims at supplying the needed mechanisms for the interoperability among heterogeneous networks and at providing meaning for sensor observations in order to enable situation

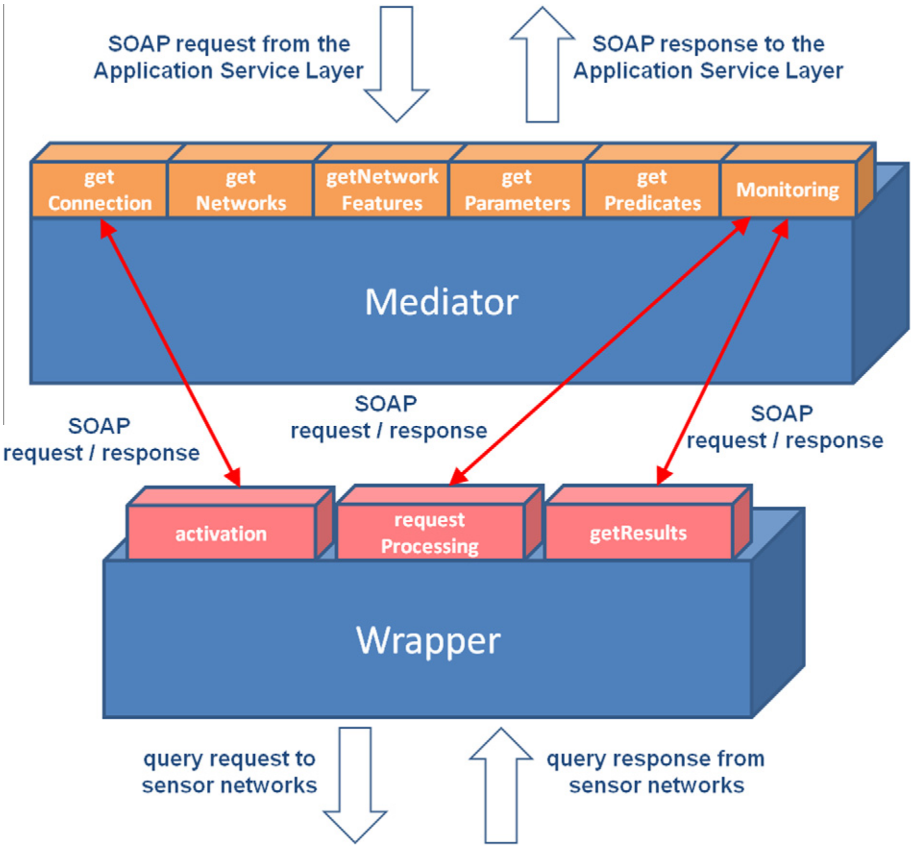

Fig. 4. Collection and integration services.

awareness. In particular, it enables any application to access heterogeneous sensor network data and to translate proprietary data format into the defined common data model. To achieve this goal different modules are needed (see Fig. 8):

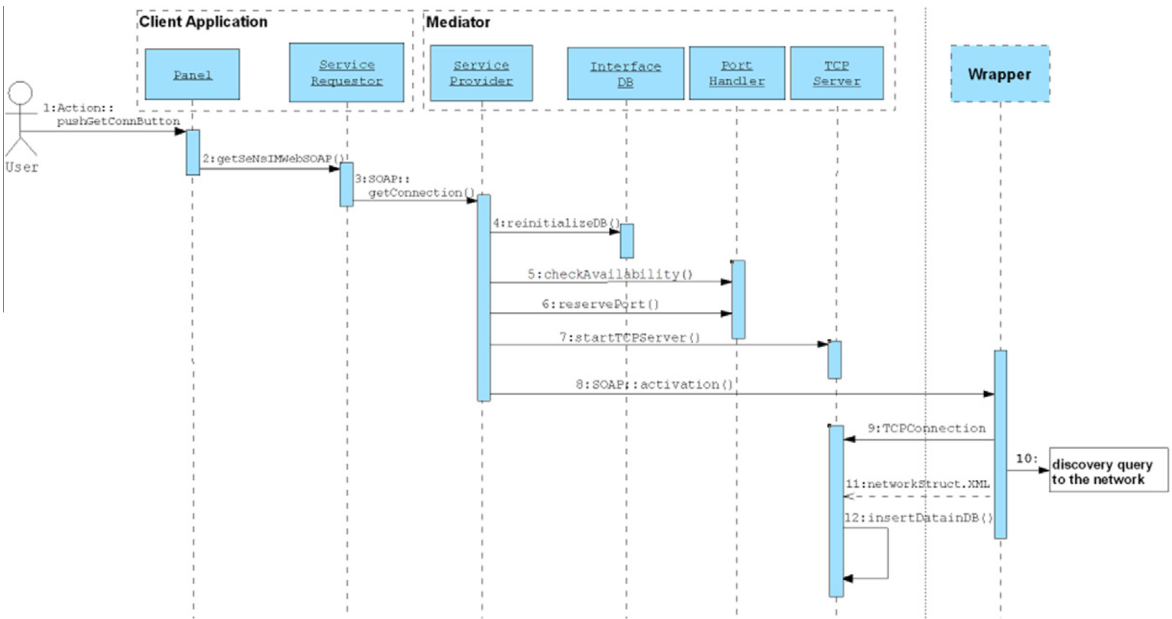

Fig. 5. *getConnection* Service.

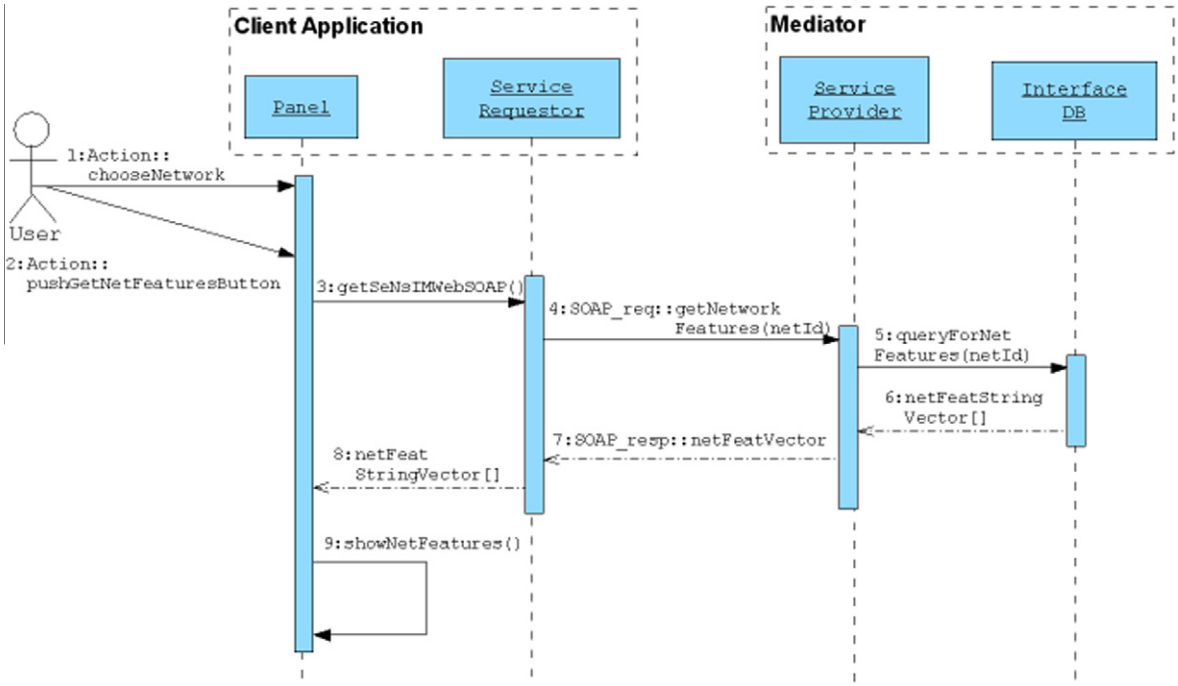

Fig. 6. *getNetworkFeatures* Service.

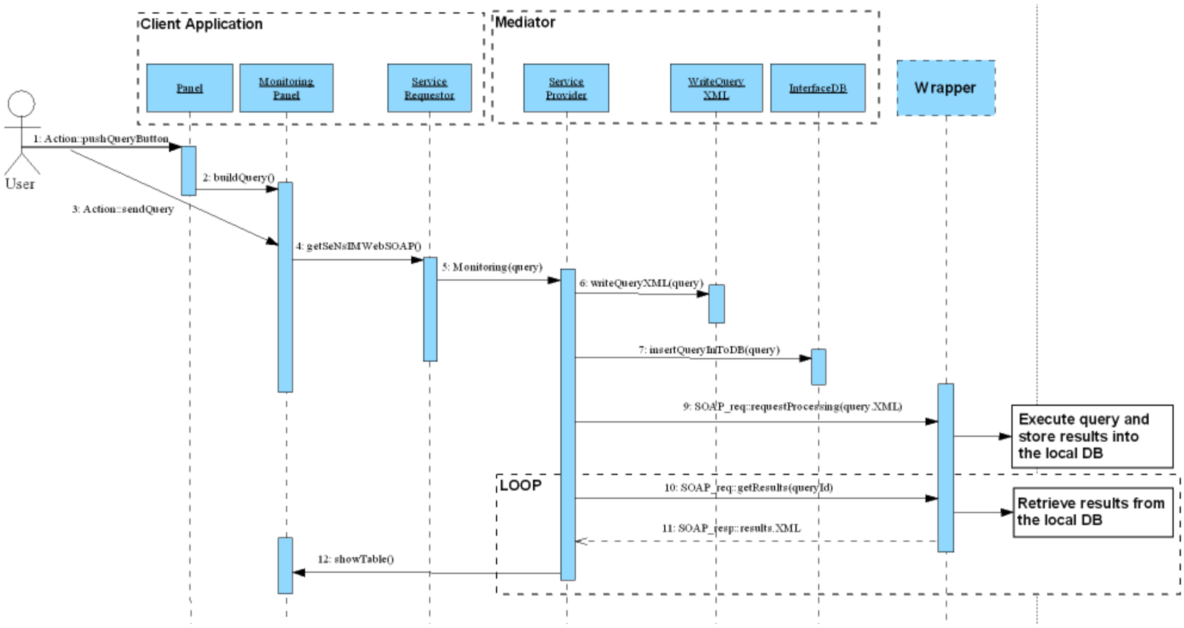

Fig. 7. *Monitoring* service.

1. Sensors add-on (data aggregator): this module is specific for each kind of sensor network. It retrieves samples from sensor systems and converts them into an XML format, according to defined translation rules.
2. XML-RDF Wrapper: this module wraps the XML data sets, translating the file into an RDF document, according to a set of sensor domain ontologies, to enrich the data with implicit semantic relations.
3. O& M Data modeler: the last module retrieves an RDF document and converts it in an O& M standard XML document.

To explain the transformation process performed by this module, we will illustrate the steps through a running example, in particular, we will consider a pluviometric sensor network to measure the rain level.

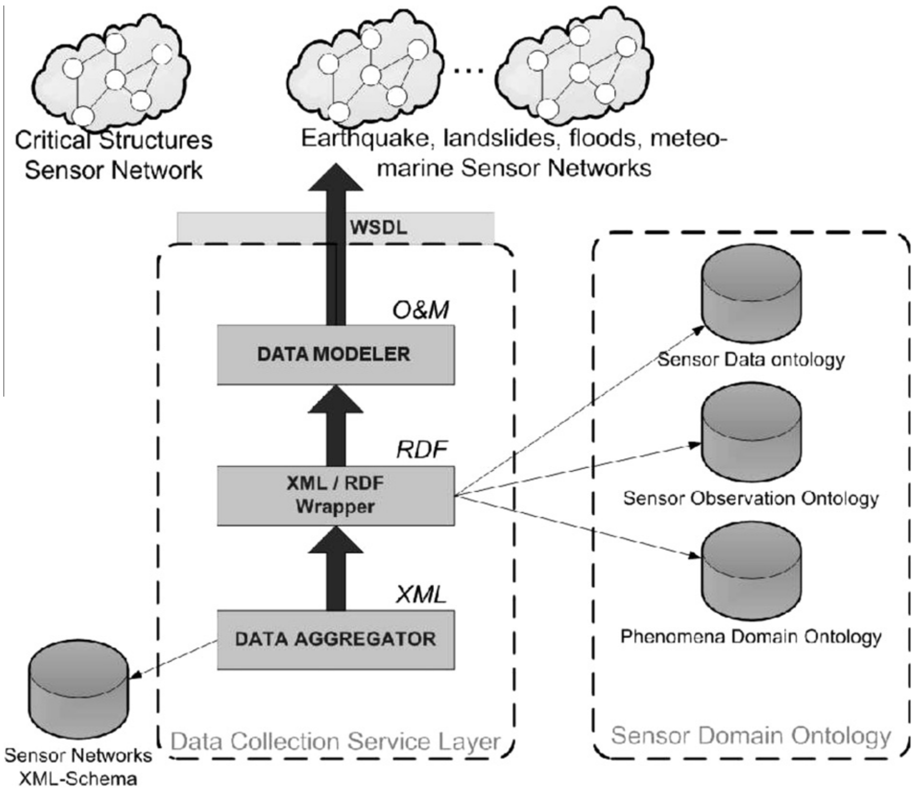

Fig. 8. The data collection layer.

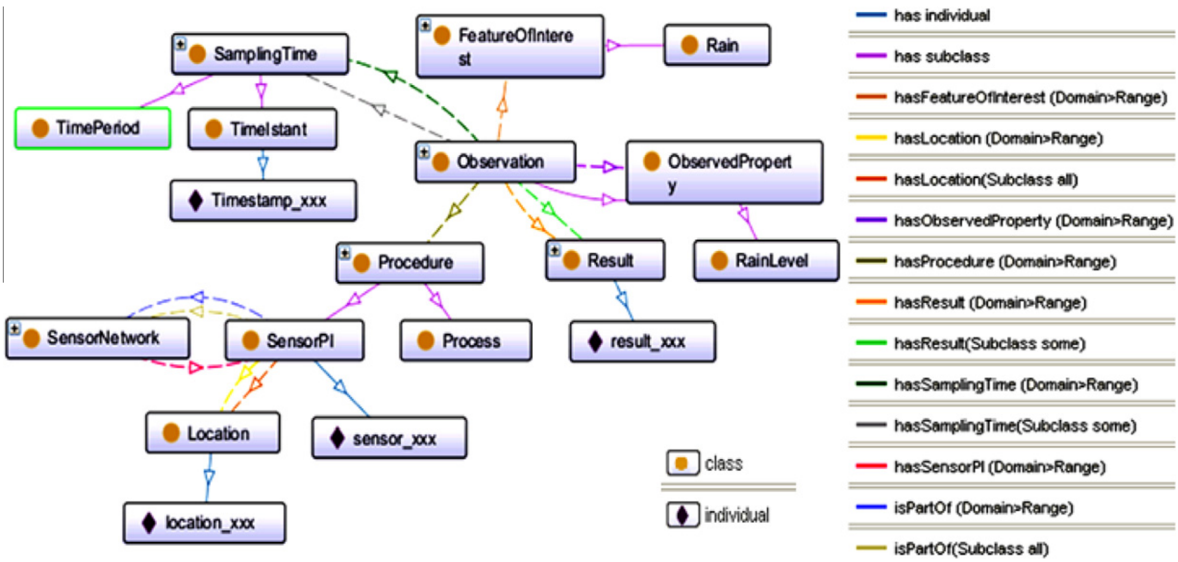

Fig. 9. Sensor domain ontology coded in RDFS schema.

The **data aggregator module** acquires raw sensor data from heterogeneous sensor networks and encodes them into an XML document. The mapping is done using an XML-schema defined for sensor networks and stored into the XML-Schema Repository.

As illustrated in Fig. 10, the XML schema is very simple; in Listing 1 there is an example of sensed data codified according to this schema; it describes the sensor coordinates, expressed according to the W3C reference Datatypes (tag origin),the sendor ID, the timestamp and the sensed rain level value expressed as a float.

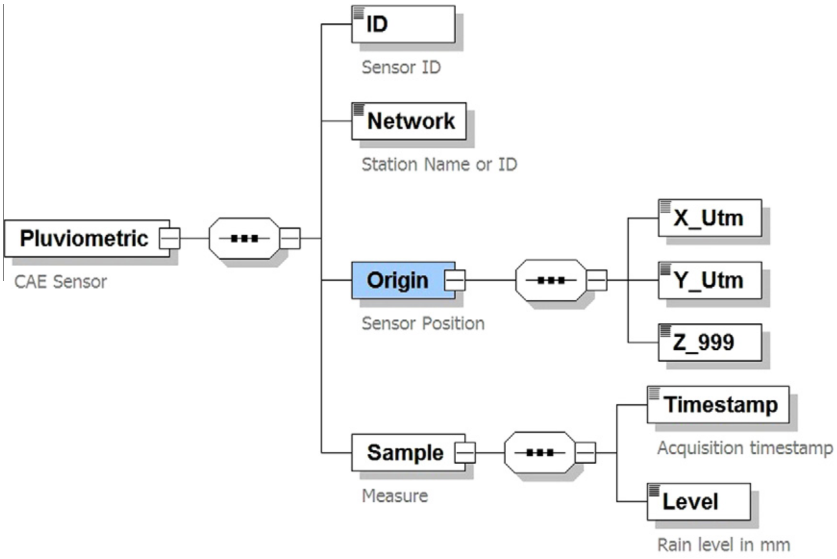

Fig. 10. XML schema for pluviometric data.

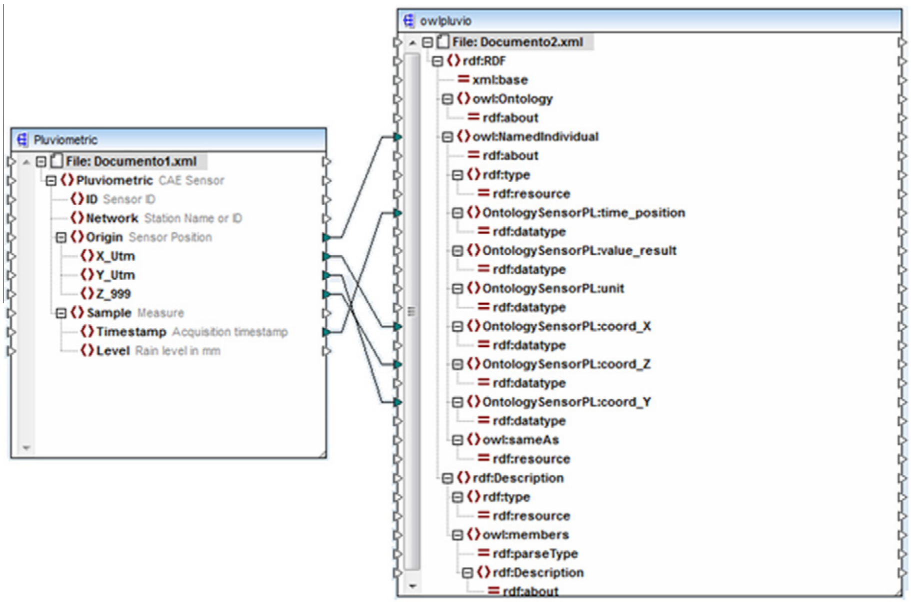

Fig. 11. XSLT Mapping between XML and RDF schema.

The whole XML schema can be downloaded from the Sensim Web Site. [2]

The **XML-RDF Wrapper** translates the XML document into an RDF using the concept defined into the Sensor Domain Ontology. The module works applying one or more XSLT according to the desired data format.

The XML coded data are then mapped in the ontological model, formalized in the RDF fragment of Listing 2. The RDF representation is more complex of the XML one, because the data are semantically enriched with information about concepts and the underlying relations.

The RDF coded sample fulfills the RDFS Schema that constitutes the model defined by domain experts, it is reported in Fig. 9. The RDFS model is designed in order to capture the implicit and explicit relations among data.

[2] http://www.seclab.unina.it/Sensim/Schema/pluviometric.xsd.

```
<Pluviometric xsi:noNamespaceSchemaLocation="Pluviometric.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ID>2147483647</ID>
    <Network>String</Network>
    <Origin>
        <X_Utm>2147483647</X_Utm>
        <Y_Utm>2147483647</Y_Utm>
        <Z_999>2147483647</Z_999>
    </Origin>
    <Sample>
        <Timestamp>2009-01-26T21:42:52</Timestamp>
        <Level>3.14159E0</Level>
    </Sample>
</Pluviometric>
```

Listing 1. Pluviometric measurement fragment coded in XML.

```
      <!-- http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#Timestamp_xxx -->
    <owl:NamedIndividual rdf:about="&OntologySensorPL;Timestamp_xxx">
        <rdf:type rdf:resource="&OntologySensorPL;TimeIstant"/>
        <time_position rdf:datatype="&xsd;dateTimeStamp">2009-01-26T21:42:52</time_position>
        <owl:sameAs rdf:resource="&OntologySensorPL;location_xxx"/>
        <owl:sameAs rdf:resource="&OntologySensorPL;observation_xxx"/>
        <owl:sameAs rdf:resource="&OntologySensorPL;result_xxx"/>
        <owl:sameAs rdf:resource="&OntologySensorPL;sensor_xxx"/>
    </owl:NamedIndividual>

    <!-- http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#location_xxx -->
    <owl:NamedIndividual rdf:about="&OntologySensorPL;location_xxx">
        <rdf:type rdf:resource="&OntologySensorPL;Location"/>
        <coord_X rdf:datatype="&xsd;long">2147483647</coord_X>
        <coord_Z rdf:datatype="&xsd;long">2147483647</coord_Z>
        <coord_Y rdf:datatype="&xsd;long">2147483647</coord_Y>
        <owl:sameAs rdf:resource="&OntologySensorPL;observation_xxx"/>
        <owl:sameAs rdf:resource="&OntologySensorPL;result_xxx"/>
        <owl:sameAs rdf:resource="&OntologySensorPL;sensor_xxx"/>
    </owl:NamedIndividual>

    <!-- http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#result_xxx -->
    <owl:NamedIndividual rdf:about="&OntologySensorPL;result_xxx">
        <rdf:type rdf:resource="&OntologySensorPL;Result"/>
        <value_result rdf:datatype="&xsd;float">3.14159E0</value_result>
        <unit rdf:datatype="&xsd;string">mm</unit>
        <owl:sameAs rdf:resource="&OntologySensorPL;sensor_xxx"/>
    </owl:NamedIndividual>
```

Listing 2. Pluviometric measurement fragment coded in RDF.

The code in Listing 2 reports the RDF instanced model; in particular *TimePosition* is an attribute of *dateTimeStamp* type, belonging to *TimeIstant* class, defined in the RDFS in otder to indicate the time instant. *Timestamp_xxx* is an instance of *TimeIstant*. *Coords_X* (*Y* or *Z*) are attributes, of Long type, of *Location* Class. *location_xxx* is an instance of *Location*. *Value_result* is attribute of float type, while *unit* is attribute of string type both of the *Result* Class. *Result_xxx* is an instance of *Result*.

Beyond the data property statements, we can note that we can express relations among instances (for example the *owl:sameAs* property indicates that two or more URI references actually refer to the same thing, i.e. they are individuals having the same *identity*).

In order to transform the XML Fragment of Pluviometric Measurement into the corresponding RDF fragment, an automatic XSL Trasformation is used. For our example the XSLT code maps the volume coordinates of pluviometric measurement into the corresponding rdf/owl description. The graphical mapping between the two files is shown in the Fig. 11, and the corresponding list of the XSLT transformation rules is reported in Appendix A.

Finally, the data are formatted according to the O& M standard thanks to the **data modeler**. The resulting data model express, in a standard way, concepts and relations enriched by semantic description. This assure interoperability among different sensor networks and enable advanced reasoning on heterogeneous sensed data.

In the Listing 3, we report the O& M fragment for the example data, fulfilling the Observation and measuraments (O&M) specification of the Sensor Web Enablement (SWE) framework aiming to standardize the entity and the relations among them to be modeled [27].

All the XML, RDF and O& M schemas can be downloaded by the web site of Sensim web site.[3]

In conclusion, we were able to add semantic annotations to an existing standard sensor web language to provide semantic descriptions and enhance the access to sensor data via standard web services. This objective has been accomplished with references to ontology concepts that provide more expressive concept descriptions. For example, by using a model reference to annotate a *sensor device* ontology enables uniform and interoperable characterization of sensor parameters regardless of different manufactures of the same type of sensor and their respective proprietary data representations/formats.

[3] http://www.seclab.unina.it/Sensim/Schema.

```xml
<om:Observation xmlns:gml="http://www.opengis.net/gml"
                xmlns:om="http://www.opengis.net/om/1.0"
                xmlns:swe="http://www.opengis.net/swe/1.0.1"
                xmlns:xlink="http://www.w3.org/1999/xlink"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="http://schemas.opengis.net/om/1.0.0/om/om.xsd ">
    <gml:description>Observation rain level with SensorPl</gml:description>
    <gml:name>Observation Test</gml:name>
    <om:samplingTime>
        <gml:TimeInstant>
            <gml:TimePeriod gml:id="tp1">
                <gml:beginPosition>2009-01-26T21:42:52</gml:beginPosition>
                <gml:endPosition>2009-01-26T21:42:52</gml:endPosition>
            </gml:TimePeriod>
        </gml:TimeInstant>
    </om:samplingTime
    <om:procedure xlink:href="http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#SensorPl"/>
    <om:observedProperty xlink:href="http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#RainLevel"/>
    <om:featureOfInterest xlink:href="http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#Rain/>
    <om:result xsi:type="gml:MeasureType" uom="urn:ogc:def:uom:OGC:mm">3.14159</om:result>
</om:Observation>
```

Listing 3. Pluviometric measurement fragment coded in O&M.

6. Conclusions and future works

In this paper we proposed an innovative architecture for the interoperability of sensor networks; it is based on web services technologies and on the definition of a common model for enabling semantic-based data management in sensor networks. The model promotes interoperability in presence of heterogeneous sensors, data and applications. The separation between data format and semantics, makes the proposed model general enough to face changes in sensors technologies, communication standards, or sensors networks topologies.The formal modeling of data also allows for events analysis and prediction, exploiting reasoning techniques on semantic relevant information. In a future work we intend to extend the implementation of such a model to take into account different phenomena and, furthermore, to design innovative decision support systems developed on the basis of information inferred with the help of the semantic model and provide new services at application level, too.

Appendix A

In the following we report the listing of the XLS Trasformation from XML to RDF fragment for the runnning example.

```
<xsl:stylesheet version="2.0"xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"xmlns:fn="http://www.w3.org/2005/
xpath-functions'' exclude-result-prefixes="xs fn">
  <xsl:output method=''xml'' encoding=''UTF-8'' indent=''yes"/>
  <xsl:template match="/">
    <RDF xmlns=" http://www.w3.org/1999/02/22-rdf-syntax-ns#"xmlns:OntologySensorPL=
            "http://www.semanticweb.org/ontologies/2010/5/10/OntologySensorPL.owl#"
            xmlns:owl="http://www.w3.org/2002/07/owl#">
      <xsl:attribute name="xsi:schemaLocation"
        namespace="http://www.w3.org/2001/XMLSchema-instance"
      select="' http://www.w3.org/1999/02/22-rdf-syntax-ns#owlpluvio.xsd'"/>
      <xsl:for-each select="*:Pluviometric[fn:namespace-uri () eq '']">
        <owl:NamedIndividual xsl:exclude-result-prefixes="owl">
            <OntologySensorPL:time_position xsl:exclude-result-prefixes="OntologySensorPL">
          <xsl:sequence select="xs:string (xs:dateTime (fn:string (*:Sample[fn:namespace-uri () eq
          '']/*:Timestamp[fn:namespace-uri () eq '']))))"/>
            </OntologySensorPL:time_position>
            <OntologySensorPL:coord_X xsl:exclude-result-prefixes="OntologySensorPL">
              <xsl:sequence select="xs:string (xs:integer (fn:string (*:Origin
              [fn:namespace-uri () eq '']/*:X_Utm[fn:namespace-uri () eq '']))))"/>
            </OntologySensorPL:coord_X>
            <OntologySensorPL:coord_Z xsl:exclude-result-prefixes="OntologySensorPL">
              <xsl:sequence select="xs:string (xs:integer (fn:string (*:Origin
              [fn:namespace-uri () eq '']/*:Z_999[fn:namespace-uri () eq '']))))"/>
            </OntologySensorPL:coord_Z>
            <OntologySensorPL:coord_Y xsl:exclude-result-prefixes="OntologySensorPL">
              <xsl:sequence select="xs:string (xs:integer (fn:string (*:Origin
```

```
            [fn:namespace-uri () eq '']/*:Y_Utm[fn:namespace-uri () eq '']))"/>
            </OntologySensorPL:coord_Y>
          </owl:NamedIndividual>
        </xsl:for-each>
      </RDF>
    </xsl:template>
</xsl:stylesheet>
```

References

[1] D. Gay, P. Levis, D. Culler, E. Brewer, nesC 1.1 Language Reference Manual, March 2003, <http://nescc.sourceforge.net/papers/nesc-ref.pdf>.
[2] D. Booth et al., Web Services Architecture, W3C Working Group, February 2004 <http://www.w3.org/TR/ws-arch>.
[3] S. Weibel, J. Kunze, C. Lagoze, M. Wolf, Dublin core metadata for resource discovery, September 1998 (RFC 2413, IETF).
[4] GML 3.1 specification, Open Geospatial Consortium, Inc., <http://portal.opengeospatial.org/>.
[5] Geographic information – Metadata, UNI EN ISO 19115:2003/Cor 1:2006, <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44361>.
[6] Open Distributed Processing, ISO/IEC 10746-1 Reference model, July 2006, <http://www.iso.org/iso/catalogue_detail.htm?csnumber=20696>.
[7] A. Sheth, M. Perry, Traveling the semantic web through space, time, and theme, IEEE Internet Computing 12 (2) (2008) 81–86.
[8] A. Sheth, C. Henson, S. Sahoo, Semantic sensor web, IEEE Internet Computing 12 (4) (2008) 78–83.
[9] X. Chu, R. Buyya, Service oriented sensor web, Sensor Networks and Configuration (2007) 51–74.
[10] C. Alegre, C. Monfort, P. Lazaridis, WIN: A new Service Oriented Architecture for risk management, <http://www.vfdb.de/riskcon/paper/InterschutzConference-WINPaper.pdf>.
[11] C. Alegre, H. Sassier, S. Pierotti, P. Lazaridis, A new geo-information architecture for risk management, Springer – Geo-Information for Disaster Management (2005) 543–550.
[12] F. Hakimpour, A. Geppert, Resolving semantic heterogeneity in schema integration, in: Proceedings of the International Conference on Formal Ontology in Information Systems, October 17–19, Ogunquit, Maine, USA, 2001, pp. 297–308.
[13] Natalya F. Noy, Semantic integration: a survey of ontology-based approaches, ACM SIGMOD Record 33 (4) (2004).
[14] M.G. Ceruti, Ontology for level-one sensor fusion and knowledge discovery, in: Knowledge Discovery and Ontologies Workshop at ECML/PKDD, Pisa, Italy, September 2004.
[15] M. Klien, A. Bernstein, Searching for services on the semantic web using process ontologies, in: 1st Semantic Web Working Symposium (SWWS-1), Stanford, CA, 2001.
[16] M. Eid, R. Liscano, A. El Saddik, A novel ontology for sensor networks data, in: CIMSA 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications La Coruna – Spain, 12–14 July, 2006.
[17] T. Kobialka, R. Buyya, C. Leckie, R. Kotagiri, A Sensor Web Middleware with Stateful Services for Heterogeneous Sensor Networks Intelligent Sensors, Sensor Networks and Information, 2007, ISSNIP 2007.
[18] Sensor Web <http://sensorweb.geomatics.ucalgary.ca/>.
[19] S. Liang, A. Coritoru, C. A Tao, Distributed geo-spatial infrastructure for smart sensor webs, Journal of Computers and Geosciences 31 (2005).
[20] S. Cox, Observations and measurements – observation schema (OGC 07-022r1), Open Geospatial Standard.
[21] A. Sheth, C. Henson, S. Sahoo, Semantic sensor web, IEEE Internet Computing (July/August) (2008) 78–83.
[22] K. Aberer, A. Gal, M. Hauswirth, K.U. Sattler, A.P. Sheth, Semantic challenges in sensor networks, in: Dagstuhl Seminar Proceedings 10042, <http://drops.dagstuhl.de/opus/volltexte/2010/2557>.
[23] V. Casola, A. Gaglione, A. Mazzeo, SeNsIM-Web: a service based architecture for sensor networks integration, in: Industrial Electronics, 2009, Proceedings of IECON '09. 35th Annual Conference of IEEE, Porto, Portugal, November 2009.
[24] V. Casola, A. Gaglione, A. Mazzeo, A reference architecture for sensor networks integration and management, in: IEEE Proceedings of GSN 2009, Oxford, UK, July 2009.
[25] W3C Architecture Domain, Extensible Markup Language (XML), <http://www.w3.org/XML/>.
[26] G. Wiederhold, Mediators in the architecture of future information systems, IEEE Computer 25 (3) (1992).
[27] OpenGIS Project Document OGC 03-022r3, Open GIS Consortium Inc., Simon Cox Editor, February 2003.