Effective Metadata Management in Federated Sensor Networks

Hoyoung Jeung[†] Sofiane Sarni[†] Ioannis Paparrizos[†] Saket Sathe[†]
Karl Aberer[†] Nicholas Dawes[‡] Thanasis G. Papaioannou[†] Michael Lehning[‡]

[†]Ecole Polytechnique Fédérale de Lausanne (EPFL), {firstname.lastname@epfl.ch}
[‡]WSL Institute for Snow and Avalanche Research (SLF), {lastname@slf.ch}

***Abstract*—As sensor networks become increasingly popular, heterogeneous sensor networks are being interconnected into *federated sensor networks* and provide huge volumes of sensor data to large user communities for a variety of applications. Effective metadata management plays a crucial role in processing and properly interpreting raw sensor measurement data, and needs to be performed in a collaborative fashion. Previous data management work has concentrated on metadata and data as two separate entities and has not provided specific support for joint real-time processing of metadata and sensor data. In this paper we propose a framework that allows effective sensor data and metadata management based on real-time metadata creation and join processing over federated sensor networks. The framework is established on three key mechanisms: (i) *distributed metadata joins* to allow streaming sensor data to be efficiently processed with their associated metadata, regardless of their location in the network, (ii) *automated metadata generation* to permit users to define monitoring conditions or operations for extracting and storing metadata from streaming sensor data, (iii) *advanced metadata search* utilizing various techniques specifically designed for sensor metadata querying and visualization. This framework is currently deployed and used as the backbone of a concrete application in environmental science and engineering, the Swiss Experiment, which runs a wide variety of measurements and experiments for environmental hazard forecasting and warning.**

I. Introduction

As the Sensor Internet starts to become reality, increasingly heterogeneous sensor networks are being interconnected into *federated sensor networks*, providing huge volumes of sensor data to large user communities for a variety of applications. In *SensorMap*[1] [1], for example, data producers may publish sensor data with their associated metadata and access control. Users can then query sensor data from anywhere in the world and generate Web-based visualizations of sensor data through a map-based interface. More recently, NASA, Cisco, and The Climate Group have launched a large-scale project, called *Planetary Skin*[2], aiming to develop an online collaborative platform to process and share data from satellite, airborne, sea- and land-based sensors around the globe. Similarly, *the Swiss Experiment*[3] [2], [3] is a collaboration of environmental science and information technology research projects, using a collaborative platform for sharing real-time sensor data across various institutions to improve environmental hazard forecasting and warning.

[1]http://www.sensormap.org/
[2]http://www.planetaryskin.org/
[3]http://www.swiss-experiment.ch/

To enable federated sensor networks, a reliable data management infrastructure is essential for sharing data over a large number of sensor networks and users. This requires state-of-the-art distributed computing, data management, and search technologies. This paper addresses one particular critical challenge among the open issues for enabling federated sensor networks: the effective management of *sensor metadata*. Sensor metadata, such as deployment location, data ownership, sensor specifications, sensor status, sensor calibrations and replacements, outlier and error information, plays a crucial role in processing and properly interpreting raw sensor measurement data. Thus, effective metadata management becomes even more of a critical issue for federated sensor networks due to the difficulties of handling heterogeneous sensor data sources and the collaborative use of these resources.

The distributed, collaborative management of metadata in scientific and engineering applications has received substantial attention in recent e-science projects. For example, metadata on data provenance [4]–[6] is one well-studied area, as data provenance is essential for repeatability of experiments, verification of experiment results, and in tracking the derivation of processed data. Research in this domain has, however, mainly been focused on off-line processing of experimental data. In contrast, as sensor networks produce real-time data and this data is consumed in real-time by applications, e.g. for monitoring of sensor deployments or in early warning applications, the metadata related to this real-time data also needs to be produced and consumed in real-time. This is the key problem addressed in this paper. Recent sensor portals for real-time publishing of sensor data have not addressed this problem in much detail, and are usually limited to very basic forms of annotational and static metadata, such as ownership and basic deployment information.

The framework introduced in this paper enables the highly dynamic, efficient, and interactive processing of sensor data and its associated metadata. The difficulty is that the processing requirements for sensor data and its metadata vary widely. Whereas metadata is highly structured and heterogeneous, low volume, and collaboratively managed, sensor data is highly homogeneous, high volume and, in many cases, automatically processed. Therefore our proposed architecture is based on a framework that employs more centralized repositories for metadata which interact with highly distributed data stream processors, processing the data from federated sensor networks. The interaction between the systems is in both directions: the framework allows each user to autonomously

obtain metadata from streaming sensor data. Such metadata is then sent, stored, and managed at a centralized metadata repository, reducing the cost and effort for the development and maintenance of metadata management systems for each user; At the same time, the metadata stored in the repository can also be transmitted back in real-time to the distributed sensor networks and joined with sensor data streams for online processing and visualization of sensor data. This bi-directional processing can be configured through a simple mechanism in real-time, rendering the framework highly dynamic.

Our framework has been developed in close collaboration with expert users from environmental science and engineering, and thus reflects central and immediate requirements on the use of federated sensor networks of the affected user community. The resulting system has been running as the backbone of the Swiss Experiment platform, a large-scale real federated sensor network. The three core mechanisms which enable effective sensor metadata management can be briefly summarized as follows:

1. Distributed Metadata Join.
Sensor data and its metadata are likely to be maintained on different physical nodes that are interconnected through the Internet. Since sensor data is generally processed with its associated metadata (e.g., processing sensor data while excluding invalid readings identified by metadata), such metadata joins should be processed across networks while facilitating efficiency.

To this end, our framework takes the following approach consisting of four steps: (i) users set up a configuration file for our data stream management system, called GSN (Global Sensor Networks[4]) [7][Section II-C]. The configuration includes the necessary information for metadata join processing, such as which data stream is processed with which metadata. (ii) Next, GSN connects to the sensor metadata repository (SMR) [2] where all metadata is stored, and pulls the metadata specified in the configuration to GSN. (iii) The fetched metadata is then dynamically joined with streaming sensor data. (iv) We also cache the metadata within GSN, in order to minimize data transmission over networks.

2. Automated Metadata Generation.
Although users can easily store their metadata in the SMR, it is essential to have an automated way to create and submit metadata to the SMR. Sensors often produce erroneous data values, thus the system needs to detect erroneous values in sensor data streams and stores the error information to the SMR in an automated manner, so that the errors will be excluded for data processing afterwards.

Our framework supports this functionality by permitting users to prescribe some monitoring conditions or operations to GSN. When the conditions are satisfied, GSN connects to the SMR and creates the necessary metadata automatically.

3. Advanced Search.
Users frequently search particular metadata to understand, analyze, and validate the associated sensor data. Querying metadata, however, becomes increasingly difficult as sensor networks are federated and hence the volume of data and metadata increases. Nevertheless, most related work [2], [8] merely supports basic search functionalities, e.g., keyword search, thus they are unable to effectively capture the attributes of sensor metadata for search.

Going beyond using simple keyword search, our SMR provides a rich set of advanced functionalities tailored for sensor metadata search: ranking search results using the PageRank algorithm, recommending pages that contain relevant metadata information to search conditions, graph visualization of the associations among metadata attributes, real-time bar and pie diagram representations, and presenting search results over maps.

The remainder of the paper is organized as follows. Section II presents our network model and the architecture of the framework. The three core mechanisms of the framework are described in Section III, Section IV, and Section V, respectively. Section VI discusses relevant studies to this work, followed by conclusions in Section VII.

II. The Framework

In this section, we first describe the architecture of our framework. We then discuss more details about two major system components of the framework.

A. Overview

Increasing use of sensor networks has resulted in the federation of multiple networks into a larger, virtual network. In the Swiss Experiment platform [2], [3], for example, various research institutes share the real-time environmental observation data and its metadata from many different local sensor networks, such that these networks can be accessed, compared, and processed as a single virtual network. In this network, each collaborator becomes a producer and a consumer of sensor data simultaneously.

To optimize the performance of applications on the federated sensor network, the framework manages sensor data and its metadata separately; each collaborator maintains the sensor data streams from its own sensor network locally, so that computationally expensive operations over data streams can be processed in a distributed fashion. In contrast, sensor metadata is managed in a centralized fashion in order to reduce the cost for system development and maintenance, since applications often share common requirements in terms of metadata management. This centralized metadata system also implies that data mining capabilities are improved by making the advanced metadata search capabilities possible as described in Section V.

Fig. 1 illustrates an example of the architecture of our framework in which three collaborators are interconnected over the Internet. In this framework, *Global Sensor Network* (GSN) manages streaming data produced from sensor deployments. GSN also contains the distributed metadata join processor [Section III] and the automated metadata generator

[4]http://gsn.sourceforge.net/

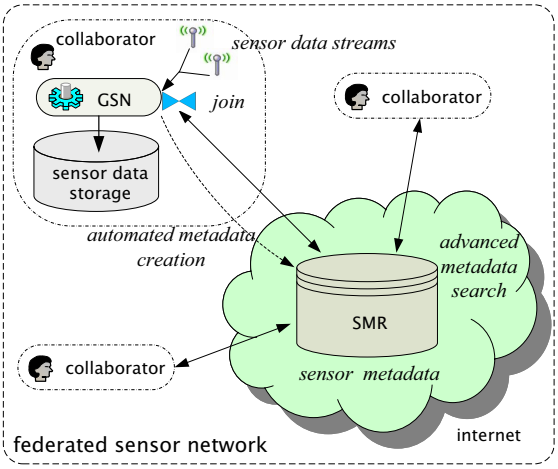

Fig. 1. Architecture of the framework.

[Section IV]. The *Sensor Metadata Repository* (SMR) manages metadata associated with the sensor data maintained by GSN. We describe more details of GSN and the SMR in the following subsections. The framework is also equipped with the advanced metadata search engine [Section IV].

Table I below provides a step-by-step description of how the framework is constructed and runs.

Step 1: GSN installation for real-time sensor data at each collaborator.
Step 2: Storing static metadata (e.g., sensor specification) to the SMR.
Step 3: Prescription of monitoring conditions to GSN.
Step 4: Dynamic metadata creation from GSN to the SMR.
Step 5: Searching both static and dynamic metadata on the SMR.

TABLE I
CONSTRUCTION OF THE FRAMEWORK.

B. Sensor Metadata Repository (SMR)

Some sensor metadata, such as sensor data type, are known a priori and are relatively static. Conversely, some metadata, such as sensor failure events, evolve over time, and thus need to be stored dynamically using the automated metadata generator described in Section IV. Fig. 2 demonstrates the metadata schema that reflects the requirements from many different scientific domains in the Swiss Experiment, covering both static and dynamic metadata in our *Sensor Metadata Repository*[5] (SMR) [2].

We store and maintain all of the sensor metadata produced by each collaborator in the SMR. The SMR is a collaborative, web-based environment where users can publish not only static metadata (e.g., specifications of sensor), but also dynamic information (e.g., sensor A is currently unavailable due to its discharged battery). In this way, users can easily submit and edit their metadata in the system without any programming, whilst the metadata join processing is performed in an automated manner and hidden from the users. Fig. 3 shows snapshots of some metadata pages in the SMR.

[5]http://lsir-swissex.epfl.ch/index.php/Fieldsite:Home

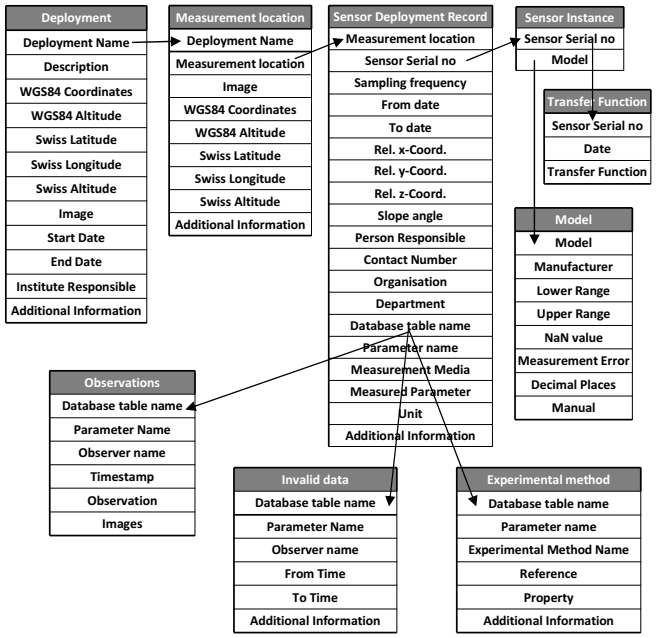

Fig. 2. Metadata schema used in the Swiss Experiment.

The SMR is built upon the *Semantic Wiki* [9] technology that simplifies the user experience, providing a user interface for entering the sensor metadata by creating a set of interlinked pages to capture the semantics of a sensor and its deployment. This offers a technique of annotating wiki pages with semantics in the form of (attribute, value)-pairs, modeling any process by meaningfully annotating the entities, and connecting them semantically to each other. For example, two users may name two sensors as of types "temperature" and "thermometer", yet want both sensors to be included in computing the average temperature of a region. The semantic Wiki approach permits the query processor to recognise that "temperature" and "thermometer" are equivalent types.

Altitude is in meters

Fieldsite HomePage	Deployment Name	Description	Altitude land surface	Organization
0168644	Mountland Visp	drought, biodiversity, land use	650	WSL
06b952e	GEDIHAP Walchwil	Sustainability, pathogen evolution, environmental health, food production, animal and plant health	420	EAWAG
070c04f	IMBALANCE Au	Biomass burning, aerosol, air quality, climate	400	ETHZ
07314c5	GEDIHAP Lumino	Sustainability, pathogen evolution, environmental health, food production, animal and plant health	270	EAWAG

Fig. 3. A snapshot of the sensor metadata repository

C. Global Sensor Network (GSN)

In the framework, each collaborator maintains streaming sensor data from its own local sensor network using our data stream management system, called *Global Sensor Network* (GSN)[6] [7]. GSN supports the flexible integration and discovery of sensor networks and sensor data, provides distributed querying and filtering, and offers the dynamic adaptation of the system configuration during operation.

In GSN, a set of wrappers allows live data to be imported into the system. The data streams are processed according to XML specification files. The middleware system is built upon a concept of sensors (real sensors or virtual sensors - new data sources created by processing or repeating live data in software) that are connected together in order to build the required processing path. For example, one can imagine an anemometer that would send its data into GSN through a wrapper (various wrappers are already available and writing new ones is quick), this data stream could then be sent to, or called by, an averaging virtual sensor using either push or pull data transfer, the output of this virtual sensor could then be split and sent to a database for recording or to a visualization layer for displaying the average measured wind in real time. GSN obtains the data directly from sensor network deployments and also provides the capability of replaying and reprocessing previously measured data.

III. Distributed Metadata Join

A wide range of applications in sensor networks commonly process streaming sensor data together with its associated metadata. For example, aggregate queries over sensor data generally exclude invalid data values caused from various sources, such as discharged batteries or network failures. As another example, suppose that a sensor was deployed at location A, then redeployed at another location B. The user may enter the metadata to record the movement of the sensor, but the logger attached to the sensor will return a single data stream, regardless of its location. If the data set does not contain positional information, it must be split into location based data sets: an operation which has previously been performed manually.

These processes can be covered by performing a join using both metadata and sensor data. In our framework, however, sensor metadata is likely to be maintained on a different server from where the corresponding sensor data is stored, as described in Subsection II-A. Thus, metadata join processing should be performed in a distributed fashion. The following subsections discuss this.

A. Join Processing

The key concept of GSN is the *virtual sensor* abstraction which enables users to declaratively specify XML-based deployment descriptors in combination with integrating sensor network data through plain SQL queries over local and remote sensor data sources. A virtual sensor generally consists of multiple wrappers, each of which allows GSN to obtain data from one of a variety of sensor data sources.

[6]http://gsn.sourceforge.net/

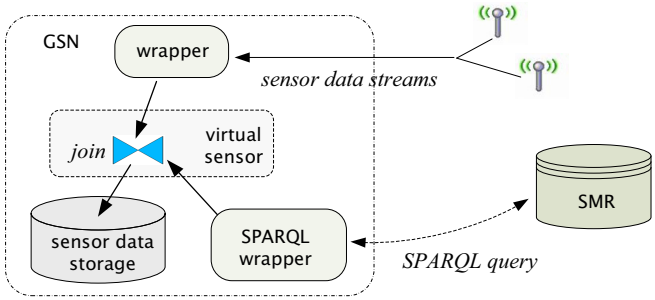

Fig. 4. Architecture of distributed join processing.

The core modules of the distributed join processing are (i) the SPARQL wrapper that communicates with the SMR to fetch necessary metadata and (ii) a special virtual sensor that performs join queries on a combination of the streaming sensor data and the metadata fetched from the SMR. Fig. 4 illustrates the architecture used for join processing in our framework.

More specifically, the join query is processed using the following steps:

1) A user composes and stores metadata into the SMR.
2) The user configures an XML-setting file for the SPARQL wrapper in GSN. It specifies the description of data acquisition from metadata, such as name of the sensor data, the associated annotation name in metadata, etc. An example of the SPARQL wrapper configuration is presented in Fig. 5.

```
<address wrapper="sparql">
   <predicate key="url">
       http://swiss-experiment.ch/sparql/
   </predicate>
   <predicate key="fields">
       Station_name:varchar(100),
       Sensor_serialno:integer,
       Project_name : varchar(100),
       Start_date: bigint,
       End_date:bigint,
   </predicate>
   <predicate key="rate">1000</predicate>
   <predicate key="query">
       <!-- SPARQL query -->
   </predicate>
</address>
```

Fig. 5. An example of configuration for SPARQL wrapper.

3) The SPARQL wrapper then dynamically composes a SPARQL query according to the configuration file. Fig. 6 provides an example of such a SPARQL query. Next, it sends the query to the SMR.

```
PREFIX a: <http://128.178.156.248/...>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?Project_name ?Station_name
        ?Sensor_serialno ?Start_date ?End_date
WHERE {
  ?page a:Property-st_name ?Station_name
  ?page a:Property-serial_no ?Sensor_serialno
  ?page a:Property-prj_name ?Project_name
  ?page a:Property-s_date ?Start_date
  ?page a:Property-e_date ?End_date
}
```

Fig. 6. An example of SPARQL query.

4) GSN caches the results from the SPARQL query, in order to increase the efficiency of the framework. Since users are likely to create or update metadata infrequently, performing SPARQL queries frequently incurs unnecessary data transmission over networks. The frequency of update for caching is configurable. Table II demonstrates an example of SPARQL query results, cached at a local server where GSN runs.

project name	station name	sensor id	start date	end date
PermaSense	position-4	2006	2007-01-02	2008-04-03
PermaSense	position-1	2006	2008-04-04	2009-09-02
PermaSense	position-2	2032	2008-08-02	2009-09-02
PermaSense	position-3	2021	2007-11-02	2009-01-06

TABLE II
AN EXAMPLE OF THE RESULTS FROM A SPARQL QUERY.

5) GSN joins the SPARQL query results from the SMR (or cached in GSN) with the corresponding sensor data streams as well as the associated operations. Fig. 7 shows a query string for the virtual sensor join.

```
SELECT s1.station_name, s1.sensor_serialno,
       s2.originatorid, s2.gentime,...
FROM s1, s2
WHERE s1.sensor_serialno = s2.originatorid
      AND s2.gentime >= s1.start_date
      AND s2.gentime <= s1.end_date
```

Fig. 7. An example of join query at virtual sensor.

B. Advantages of Distributed Join

Our approach to distributed join in the framework offers various advantages, briefly summarized as follows:

- The tasks to be carried out by users (i.e., metadata composition and setting the SPARQL wrapper configuration file) are simplified, hiding the underlying complicated join processing from users.
- The join processing is highly dynamic. As soon as users update metadata or the configuration file, the changes are reflected in the join processing instantly. The join processing is also performed in real time, whenever a new sensor reading is streamed to the system.
- The distributed join processing renders our frameworks scalable and robust because the computationally expensive join processing is distributed over collaborators.
- The approach permits collaborative work and eliminates duplicated effort among the collaborators when developing and maintaining individual metadata management systems.

IV. AUTOMATED METADATA GENERATION

In a broad range of sensor network applications, it is often necessary to monitor every data point of streaming sensor data. For example, real-time detection of sensor or network failures is important in surveillance-based applications. Monitoring data quality is high on the priority list for most environmental scientists. Nevertheless, as network sizes increase it becomes infeasible for users to monitor the large number of sensor data streams manually.

To address this, we introduce a mechanism that monitors data streams maintained in GSN, and records the monitoring results as metadata into the SMR in an automated manner. More specifically, users can define GSN's monitoring conditions or operations through a configuration file. GSN then continuously monitors whether streaming sensor data meets the conditions specified. When the conditions are satisfied, GSN connects to the SMR through the Internet and creates the necessary metadata automatically. Users can then either visualize the metadata generated in the SMR, or configure the framework to send notifications of events (e.g., SMS).

In the following subsections, we describe a complete example of such automated metdata generation, dealing with quality of streaming sensor data.

A. Dynamic Quality Measure

Sensor measurements are often corrupted by external environmental parameters, such as freezing or heating of the casing or measurement device, accumulation of dirt, mechanical failure or vandalism (from humans or animals). To cope with this, we have been developing a mechanism that measures the quality of sensor data, based on various mathematical models. The mechanism includes visualization of data quality associated with the raw data, which could greatly improve the users' understanding of the data. Suppose that a raw data stream is plotted as a graph, which is then overlaid by another graph showing different colors or a histogram to indicate detected "dirty" data points. Users may then manually check whether the detected data points are realistic (by visual observation). In this way, users would be able to verify the data quality measure.

The process of sensor metadata generation is illustrated in Fig. 8. For all raw values transmitted by sensors, we associate a quality value. This information indicates the relevance of a particular sensor value with respect to errors which could have occurred while sensing or transmitting data.

Dynamic metadata generation with respect to data quality poses important challenges. For example, consider a temperature sensor in a deployment handled by GSN. If the data

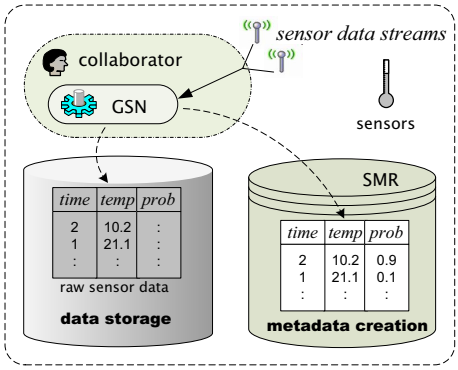

Fig. 8. Example of metadata creation system.

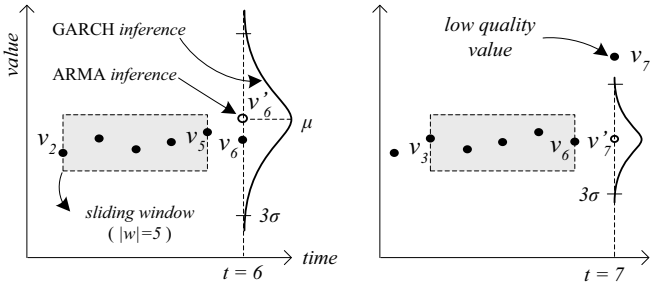

Fig. 9. An example of nonparametric diagnostics.

quality measure is inappropriate, then erroneous data would be used for data processing, which may yield incorrect results.

To address this, we employ a probabilistic approach for computing the quality measure in order to capture significant changes in a data stream using probabilistic models. One of these probabilistic models is described in the following subsection. It applies to environmental parameters which cannot physically change rapidly in the short term. Thus, if a sensor suddenly reports dramatic and unusual changes, then we can say that the quality of values produced is poor. We use this observation for generating quality related metadata.

B. Probabilistic Quality Metric

One example of an established quality control on normally distributed data is presented here. We note that a suitable automated quality control depends on the type of data. For such normally distributed data, each data stream is modelled to have a Gaussian probability distribution at each time t. This is carried out as follows: for each raw value v_t at time t, we assume that it contains some white noise a_t, which is modeled as $v_t = v'_t + a_t$, where v'_t denotes an unobservable true value. Given a (sliding) window $w = \langle v_{t-|w|-1},\ v_{t-|w|}, \cdots, v_{t-1} \rangle$ having $|w|$ values, we then infer v'_t and a_t using the ARMA (AutoRegressive Moving Average) model [10].

Next, we use these a_t to estimate volatility (σ_t) of the time series using the GARCH (Generalized AutoRegressive Conditional Heteroskedasticity) model [11]. Thus the GARCH model computes time-varying volatilities using a_t. Moreover, by using ARMA and GARCH models we infer a Gaussian probability distribution, $N(v'_t, \sigma_t)$, at each time step t.

Next, we use the inferred Gaussian probability distribution for generating metadata related to quality information. Let $Q(v_t)$ denote the probability of v_t occurring *w.r.t* $N(v'_t, \sigma_t)$,

$$Q(v_t) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{(v_t - v'_t)^2}{2\sigma_t^2}\right). \quad (1)$$

Intuitively, $Q(v_t)$ gives us the likelihood of observing v_t given recent values in a window w. A low value of $Q(v_t)$ means that it is very unlikely that we observed a value like v_t and thus conclude that the quality of v_t is low. On the other hand, if $Q(v_t)$ is high then we can conclude that the quality of v_t is high. Thus, $Q(v_t)$ can be used for denoting quality information to raw sensors values.

Fig. 9 shows an example of metadata creation. At time $t = 6$, the ARMA model infers the expected true value v'_6 using the values $v_2, \cdots, v_5$ in the sliding window, which is served as the mean in Gaussian distribution. Likewise, the GARCH model infers the variance at $t = 6$. Thus, the Gaussian distribution at $t = 6$ is formed by the inferences of ARMA and GARCH models. Next, we compute the quality information of the raw value v_6 by using the quality measure from (1) which clearly indicates that v_6 is of a high quality (i.e., v_6 is close to its corresponding expected true value v'_6). At $t = 7$, the window has been slid and contains values from v_3 to v_6. We repeat the same process as the previous one at $t = 6$, and find v_7 has a low quality measure and thus could be considered dirty or erroneous. Thus, the metadata creation process gives a lower quality value to v_7. Note that the inferred densities at $t = 6$ and at $t = 7$ should not be identical since we compute time-dependent probability densities by taking different values for the sliding window at each time.

V. Advanced Metadata Search

In sensor network applications, users often search particular metadata to understand, analyze, and validate associated sensor data. Querying metadata, however, may become difficult across federated sensor networks, since the volume of metadata is likely to rise very quickly.

The semantic MediaWiki, i.e., the backbone of our sensor metadata repository (SMR), has the capability of performing a basic keyword search using the SPARQL [12] query language. Although SPARQL is originally designed for RDF data, it suits the SMR, since the semantic MediaWiki has the capability of storing the data in the form of RDF graphs.

Nevertheless, this basic search functionality is unable to effectively capture the attributes of sensor metadata for search. For example, we cannot query metadata as "*report all sensors deployed in 2009 at region A*" or "*display all our sensors on a map*". Moreover, keyword search may retrive too much relevant information (i.e., metadata pages) if the database of metadata is large.

In this section, we introduce an advanced search system for sensor metadata, built upon the SMR, yet equipped with a rich set of technologies that can effectively help users search sensor metadata and understand the results.

A. System Overview

User Interface: The system provides an easy-to-use query interface that takes user's inputs for queries and demonstrates the results with various visualization tools. It is designed for users with no prior knowledge of metadata stored in the system. A value for searching the corresponding property is selected using a dropdown list whose values are dynamically created by probing the metadata database. The query form also allows users to add multiple properties to the query.

In addition to taking users' inputs for search, the query interface provides several formats for displaying search results, such as type of visualization (e.g., bar, pie, graph, and calendar) and aggregation of query results (e.g., sum, min, max and count). Users can also export the search results into XML or CSV (comma separated values) files.

Search Engine: Fig. 10 illustrates the search mechanisms in our system. *Query Management* is responsible for processing queries, whilst taking into consideration the mapping of RDF schema to database schema. It also connects with several other modules—the Google Maps API, the GraphViz library, the Google Pie and Bar APIs, and the Calendar API—for dynamically visualizing search results in effective formats.

The results matched to queries are ranked by the PageRank algorithm [13]. This becomes very useful when a search results in many metadata pages, because the PageRank algorithm allows more popular metadata pages to be shown at the top of the result form. In the SMR, every metadata page has two kinds of linking structures: one is the links provided by the RDF graphs and metadata properties, and the other is normal web-page links from one another. We extend the original PageRank algorithm to consider these two links simultaneously for scoring the metadata pages. The following table summarizes some statistics of the pages in the SMR, used for the PageRank algorithm.

number of pages in the SMR	20515
number of links (edges) among the pages	587210
number of gangling nodes	15998
maximum in-degree	4517
maximum out-degree	1300

In addition to ranking the pages satisfying search conditions, a recommendation mechanism is embedded into our search engine. It presents relevant pages based on the combination of query inputs and properties, which have high scores in the PageRank algorithm. This function is useful when a search result set has only few metadata pages to show, since users can view relevant metadata information in addition to the (direct) search results.

Visualization of Search Results: The advanced search displays results in various formats, according to the settings users make through the query interface. These include maps, graphs, bar and pie diagrams, in addition to plain tabular formats to demonstrate the results. The visualization tools provided by the advanced search displays help users intuitively understand query results. We briefly describe two important tools for displaying results. In the map visualization, any query results containing positional information are presented with

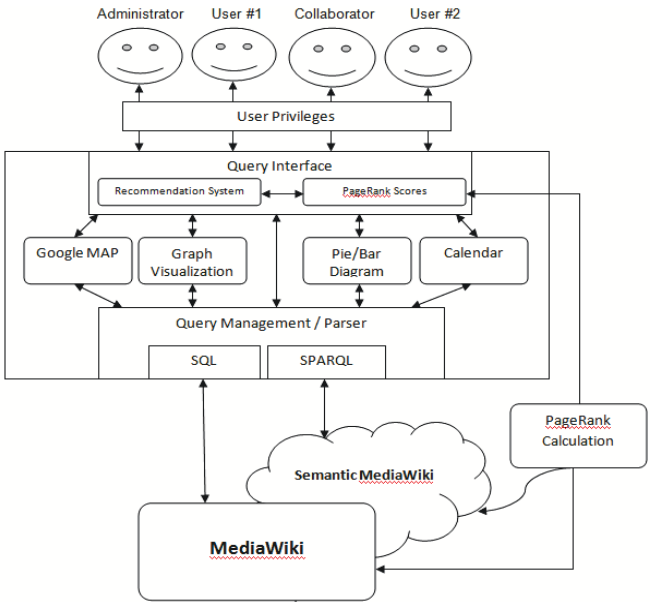

Fig. 10. Architecture of query processor.

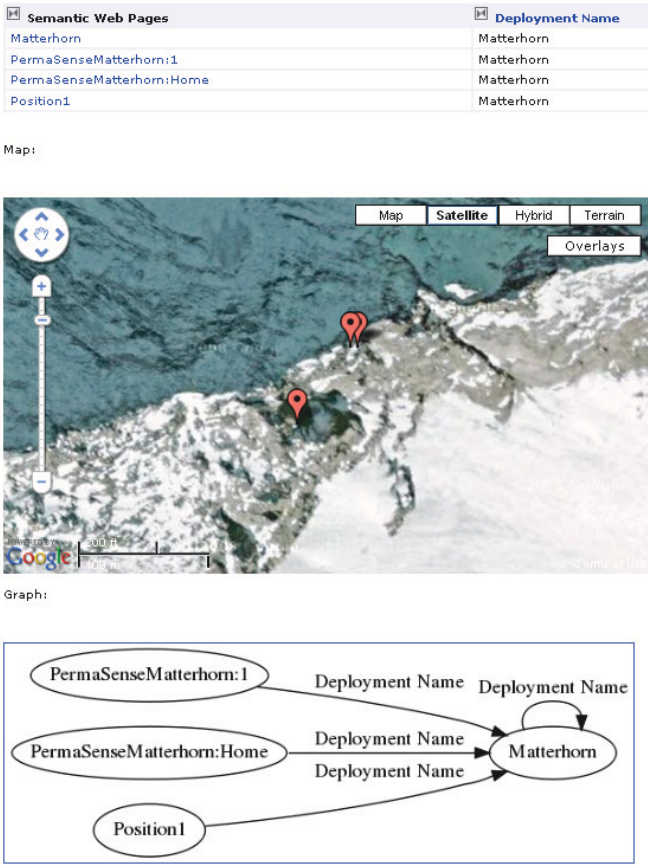

Fig. 11. A snapshot of search results.

various colors on the Google Maps, as well as the links to the corresponding metadata pages in a list form under the map. The graph visualization represents the associations of sensor metadata in the results. Each metadata page may have references to several properties that may be either identical or different from each other. Thus, this module classifies the metadata using the similarities of their properties. Then, the directed arcs (links) are used to show the associations. Fig. 11 demonstrates snapshots of the graph and the map visualizations

VI. Related Work

A rich body of previous work on managing metadata exits. Data provenance [5], [6], [14] has been a hot research topic recently and is finding its application in a variety of research problems [15], [16], especially in e-science [17], [18].

Our previous work [2] considers the requirement for data provenance in a collaborative environment where users publish raw sensor data in the form of virtual sensors and post-process data by means of filtering, modeling, or query processing techniques. This work has been developed and generalized, the results of which were shown in Fig 2. Data from different sources with different provenance is enriched with further metadata at each processing step to describe the processing implemented and/or observations which may explain anomalies in the data.

Data annotation is a term, often used in relation to data provenance. Usually, researchers not only produce and consume data, but they also comment on it and refer to it, as well as referring to the results of queries upon it. Annotation is therefore a significant aspect of scientific communication. One researcher may wish to highlight a point in data space for another to investigate further. They may wish to annotate the result of a query such that similar queries show the annotation. J. Zhao et al. [4] carried out research into annotating, linking and browsing provenance logs for e-science using a conceptual open hypermedia system to build a dynamically generated hypertext of web of provenance documents. Their work does not, however, deal with the acquisition of annotations and metadata storage which is given more weight in this paper.

Fox [8] discusses different sources of metadata and approaches to metadata. However the term "metadata" is loosely defined and has been used in variety of contexts. In addition, much research has been carried out in the area of data annotation, but less effort has been put into documentation of the metadata and its storage in the e-science context.

Several e-science projects[7] [8] [9] make use of Web 2.0 portals in the form of wikis or custom made environments to foster collaboration among scientists. For instance, Kepler and our own platform [2], [3] are based on the MediaWiki platform and employ several extensions to ease its use. Our project, however, aims to provide support for all types of metadata, whereas the other projects concentrate only on workflows.

[7]http://datafedwiki.wustl.edu/index.php/DataFed/

[8]http://wiki.myexperiment.org/index.php/Main Page/

[9]http://kepler-project.org/

VII. Conclusions

The explosive increase in sensor network use in a variety of domains has resulted in the emergence of applications built on federated sensor networks. As applications typically produce and share huge amounts of data from sensors, managing metadata as well as sensor data becomes important. This paper has introduced an effective framework for managing sensor metadata whilst considering real-time metadata creation and processing over distributed collaborators. The framework is enabled by three primary mechanisms: *distributed metadata joins* with streaming sensor data, *automated metadata generation* from sensor data streams, and *advanced metadata searches* based on various techniques for querying and visualizing sensor metadata.

The systems developed in the framework are not only interesting and visually enticing for non-expert users but bring substantial benefits to applications on federated sensor networks. The cutting-edge technologies developed for sensor metadata management will open up new ground in collaborative data gathering and interpretation.

References

[1] Y. Ahmad and S. Nath, "COLR-Tree: Communication-efficient spatio-temporal indexing for a sensor data web portal," in *ICDE*, 2008, pp. 784–793.

[2] N. Dawes, K. A. Kumar, S. Michel, K. Aberer, and M. Lehning, "Sensor metadata management and its application in collaborative environmental research," in *eScience*, 2008, pp. 143–150.

[3] S. Michel, A. Salehi, L. Luo, N. Dawes, K. Aberer, G. Barrenetxea, M. Bavay, A. Kansal, K. A. Kumar, S. Nath, M. Parlange, S. Tansley, C. van Ingen, F. Zhao, and Y. Zhou, "Environmental monitoring 2.0," in *ICDE*, 2009, pp. 1507–1510.

[4] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer, "Semantically linking and browsing provenance logs for e-science," in *ICSNW*, 2004, pp. 158–176.

[5] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Record*, vol. 34, no. 3, p. 3136, 2005.

[6] R. S. Barga and L. A. Digiampietri, "Automatic generation of workflow provenance," in *IPAW*, 2006, pp. 1–9.

[7] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *VLDB*, 2006, pp. 1199–1202.

[8] G. Fox, "Data and metadata on the semantic grid," *Computing in Science and Engineering*, vol. 5, no. 5, pp. 76–78, 2003.

[9] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer, "Semantic wikipedia," in *WWW*, 2006, pp. 585–594.

[10] R. Shumway and D. Stoffer, *Time Series Analysis and Its Applications (Springer Texts in Statistics)*. Springer-Verlag, New York, 2005.

[11] T. Bollerslev, "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics*, vol. 31, pp. 307–327, 1986.

[12] "SPARQL query language for RDF," W3C Recommendation, World Wide Web Consortium, Tech. Rep., January 2008.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report, 1999.

[14] B. Glavic and K. R. Dittrich, "Data provenance: A categorization of existing approaches," in *BTW*, 2007, pp. 227–241.

[15] P. Buneman and W. C. Tan, "Provenance in databases," in *SIGMOD*, 2007, p. 11711173.

[16] S. Miles, S. Munroe, M. Luck, and L. Moreau, "Modelling the provenance of data in autonomous systems," in *AAMAS*, 2007, p. 50.

[17] S. Miles, S. C. Wong, W. Fang, P. Groth, K.-P. Zauner, and L. Moreau, "Provenance-based validation of e-science experiments," *Web Semant.*, vol. 5, no. 1, pp. 28–38, 2007.

[18] S. B. Davidson, S. C. Boulakia, A. Eyal, B. Ludascher, T. M. McPhillips, S. Bowers, M. K. Anand, and J. Freire, "Provenance in scientific workflow systems," *IEEE Data Eng. Bull.*, vol. 30, no. 4, pp. 44–50, 2007.