SensorFeed: An Architecture for Model-based Sensor Network Data Enrichment

Ali Salehi, Mukaddim Pathan, Doug Palmer, and Michael Compton

Information Engineering Laboratory, CSIRO ICT Centre
108 North Road, Acton, ACT 2601, Australia
{ali.salehi, mukaddim.pathan, doug.palmer, michael.compton}@csiro.au

***Abstract*—Perceiving the environment, a sensor network collects huge volume of data to be used in different application domains. The gathered data is often used and analyzed by e-research scientists other than the original investigator. Therefore, the sensed data needs to be captured, processed and stored in a form that will allow someone to use the data with confidence long after the original investigators have left the scene. To address this need, in this paper we present SensorFeed, which integrates metadata repositories and sensor data management systems. Using SensorFeed, scientists can annotate sensor readings automatically as they are streamed, through direct use of statistical modeling frameworks. These annotations enrich sensor readings, thus making datasets generated from sensor network deployments usable by external scientists. A real-world use case of SensorFeed for agriculture engineering is presented to show the applicability of our approach to an e-research application.**

I. INTRODUCTION

Sensor networks [3, 30], inherently heterogeneous in nature, are distributed across the globe. They produce a large amount of dense, in-situ data about an environment. Since the sensor network data provides a snapshot of the conditions in the network boundaries, it can be re-used and re-analyzed by scientists other than the original investigators. Collected data from heterogeneous sensor networks is often archived or streamed as raw data, but it has to be associated with metadata, describing its meaning. The underlying reasons for enhanced meaning of sensor data are to enable situation awareness, ongoing citation and access, unaffected by changes in storage and services that the data might undergo over its lifespan.

Metadata annotation, i.e. providing meaning to sensor data, includes the feature of interest, the specification of measuring instruments, accuracy, location, condition, and scenario of measurements. Creating metadata and keeping it up-to-date and usable is part of the strategy for the long-term sensor network data management for e-research [4]. It is especially important as environmental scientists use sensor network deployments for detailed monitoring of the physical world and these deployments generate unprecedented amount of data. Managing the data effectively is essential to support the full lifecycle of an e-research endeavor, from concept formulation and outlining of the research activity itself, to data collection, processing, metadata annotation, provenance, curation, discovery, analysis, and dissemination of research results. Moreover, metadata annotation is essential when a user is confronted with large numbers of sensors and sheer volume of collected data. When it is not clear what is available, a user can start a general search of relevant concepts and narrow it down based on semantic descriptions and their relations.

In the context of sensor data management [31], specifically in the e-research domain, there is still a large gap between data collection and metadata annotation. The sharing of data within the e-research community is of little significance without effective metadata management both at the source and throughout the processing chain. Without appropriate methods for management, processing, analysis, metadata generation and annotation, collected sensor data may be misinterpreted by a third party. It is for this reason that e-researchers have traditionally preferred to collect their own data.

From an e-research application perspective, the collected data from a sensor network needs to be managed to remain accessible, discoverable, and reusable over a long period of time. Data management does not happen automatically, and it cannot be the responsibility of an e-researcher alone. Rather there should be an appropriate data management plan, as a partnership, between sensor network investigators and e-researchers, to share the burden of nurturing and maintaining sensor data for the wider community. Although there are some data management systems [1, 5, 9, 22] available for life scientists, they are mostly domain specific and not quite usable for cross-domain research and collaboration.

In this paper, we endeavor to break through these barriers by presenting *SensorFeed*, a sensor network data management middleware, which integrates metadata repositories and sensor data management systems. It makes use of statistical modeling frameworks to automatically annotate sensor readings and stores in a form that allows external scientists to use, discover, and re-use them, even after a long period of time. The main contributions of this paper are:

- A model-based architectural description of a middleware to perform data management for e-research applications.
- An e-research case study to demonstrate the applicability of the proposed data management middleware

The rest of the paper is structured as follows. Section II describes the need for a model-based sensor data management middleware for e-research. It is followed by a comparative analysis of existing research works. The proposed architecture and associated data structure are presented in Section IV and Section V, respectively, followed by a case study. The paper is concluded in Section VII.

II. MOTIVATION

E-research [4] revolutionizes every aspect of scientific practice in the fields of medicine, genetics, chemistry, education, linguistics, and finance. It makes use of distributed sensor networks, high performance computing infrastructure, data sources, scientific instruments, and communications technologies to enable scientists to perform their research independent of time and geographic location.

Once data is collected from sensor networks involving thousands of sensor nodes, multi-disciplinary research teams are often engaged in accessing, sharing, and processing those data stored in digital repositories. Therefore, the collected data is to be associated with appropriate meaning to be usable and understandable to all parties. The data from a sensor network is very different to the data contained in, for example, a radio telescope image. Rather than a large quantity of data that can be represented by common metadata, sensor network data streams—and occasional individual values—may have large collections of metadata associated with them.

There are a number of catalogues of metadata, such as Australian National Data Service (ANDS), AuScope, and INSPIRE that allow researchers to locate datasets and services that can provide relevant information. In addition, different users, disciplines and applications have built their own data management solutions. With the growth of collaborative e-research and applications, it has become evident that this individual approach can be wasteful, leading to expensive duplication of resources and lack of interoperability between developed solutions. In addition, e-researchers are often not familiar with data models or concepts such as metadata, databases, data enrichment, or query languages. To them data are mostly observations from sensor networks, involving the actual measurements, their knowledge about the field site, the deployment and execution of the scientific experiment [29]. In order to address these issues, a middleware needs to be in place to perform automatic annotation of sensor readings as they are streamed. The middleware can provide a common set of services and tools to allow e-researchers and their applications to treat digital repositories, data sources, sensor network and computing infrastructure, and other disparate resources as one large virtual facility.

Our research proceeds in this direction with the development of *SensorFeed*, a sensor data management middleware that allows e-researchers to perform experiments and share their observations and results in real time. It endeavors to define a layer of Application Programming Interfaces (APIs) and systems to isolate the users and their applications from any particular characteristics of data sources and resources that they access. This means that scientists can access the same or a collection of heterogeneous resources without making any special arrangement on a per resource basis or by altering a particular application.

III. Related Work

Data stream management systems (DSMS) are among some of the most studied research subjects recently. These systems are designed to provide quick response time when dealing with large volumes of data, e.g. sensor observations. These systems employ window-based data processing combined with synopsis to process large volumes of data [6, 14, 20]. Using synopsis helps a DSMS in reducing the response time to queries. Global Sensor Network (GSN) [2], TelegraphCQ [9], Aurora [1] and Stream [5] are among some of the known works in this domain. There are also Internet-based streaming systems, such as Stream-based Overlay Network (SBON) [23] and Peer-to-peer Information Exchange and Retrieval (PIER) [15] that process and deliver data over the Internet. They rely on P2P model for data representation, query dissemination, operations and metadata management. These systems are appealing since they address the challenges related to large scale sensor resource and data sharing.

There exist research projects to provide access, query, streaming, and management of sensor network data. The Sensor Web project [11] provides a dynamic infrastructure that allows users to access sensor networks and stream data out. Sensor Information Networking Architecture (SINA) [25] is a middleware for querying, monitoring, and tasking of sensor networks. Tiny Application Sensor Kit (TASK) [8] is built on top of TinyDB to provide high level metadata management, query configuration, monitoring and data visualization. Many sensor network applications are built on homogeneous architecture, thus they suffer from the lack of interoperability and also cannot provide unified services. To address these issues, like several other research projects in this field [10, 16, 19], we intend to build SensorFeed on top of the GSN platform. Using GSN, internal details of the sensor networks are abstracted from the application-domain code, hence more flexibility and higher interoperability are obtained.

Research projects such as BIRN [12], Kepler [18] and Taverna [22] provide workflow-based data management infrastructure. In addition, there are metadata standards such as Darwin Core [28], Sensor Web Enablement (SWE) [7], Data Documentation Initiative (DDI) [27] and Directory Interchange Format (DIF) [21]. However, analyses of existing literature reveal that only limited number of data stream and metadata management systems are usable by non-computer scientists, e.g. hydrologists, biologists, botanists, geneticists, geologists or medical scientists. In the works [10, 16], authors present a multi-tiered metadata management system that uses GSN as the backend for sensor data management and a semantic wiki engine to handle metadata information. Authors in these works use a SPARQL query language [24] to link GSN to the semantic wiki's storage engine. While these works are among the cornerstones in integrating sensor data and metadata repositories, they are not replicable by other researchers due to the tight integration of the underlying systems with the actual local use-cases. Although our focus in this paper is on plant research, with SensorFeed, we attempt to build a highly flexible, domain-independent sensor data and metadata management system, which is tightly coupled with statistical modeling environments.

IV. E-Research Data Management

A sensor network potentially can have thousands of nodes and sensors, with each sensor producing a semi-independent time series. For the purposes of data management for e-research, a decentralized approach can be followed.

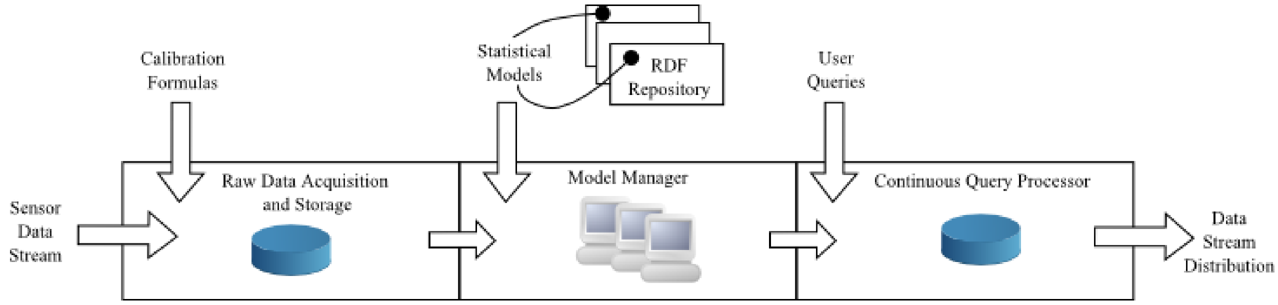

Fig. 1. SensorFeed architecture

A. Data Enrichment

The number of data streams is typically large for manual handling, thus requiring machine-to-machine processing. We call this process *enrichment*, involving the following steps:

1) Collection: Data streams are collected from a sensor network and are associated with sensors, nodes, and positions.

2) Processing: The collected data is processed into physical measurements and combined into composites, e.g. combining moisture sensor readings into a moisture map.

3) Metadata Annotation: Processed data and streams are associated with meaning, confidence and quality information.

4) Provenance: The data is associated with information on how it was processed (derivation), for whom and why it was collected (agency), and how it may be distributed (rights).

5) Curation: The data is validated and stored in digital repositories or virtual observatories with information on how it can be accessed. Links to similar datasets can also be added.

6) Discovery: The annotated metadata is presented in a form that allows semantic search and query. Thinned datasets may be produced to allow exploration.

7) Analysis: The datasets are imported into analysis tools and modeling is performed for use in e-research applications.

The operations along the enrichment hierarchy depend on suitable storage and middleware technologies. In our work, we provide a flexible architecture using which scientists can construct and automate the aforementioned process.

B. Metadata Management

As presented in earlier, metadata not only adds value to the raw sensor readings, e.g. metadata for quality of the sensor measurements, but also enables reuse of sensor data by other scientists. Metadata is usually provided by scientists manually. They use semantically enabled systems, e.g. metadata registry, to enter the relevant metadata in the form of tags, concepts and relationships. Internally, metadata is presented using the Resource Description Framework (RDF) model. Metadata information is then associated with sensor readings.

The process of associating metadata with sensor readings can be done either manually or automatically. In the manual form, scientists can link either a single data point or a sequence of data points with one or more metadata instances. In automated annotation, scientists use statistical models to analyze the dataset and dynamically link data points or a range of data to one or more RDF data models. Automated annotation is complementary to manual annotation.

As an example of manual annotation, consider a scientist visits a weather station and finds that a wind sensor is broken. He/she manually marks the dataset with an invalid annotation. This annotation communicates the existence of a broken sensor to other scientists who may reuse this dataset in future.

As an example for automated annotation, consider a model-based data cleaning system in which statistical models are used to grade quality of sensor readings as they are streamed into a data management system. These models are normally executed inside statistical environments such as R or Matlab. A statistical model could range from a combination of simple moving averages to complex clustering algorithms with multiple data transformations. While supporting manual annotation is essential for scientists, automated annotation is the key to enable real-time data sharing and reuse. This is due to the fact that, to maximize usability of a data point, it has to come with relevant metadata such as location, hardware specification, and calibration factors, among others.

C. Proposed Architecture

To support the vision of having a middleware that enables e-researchers to conduct experiments, and share observations and results in real-time, we have to tackle three key challenges. First, a system architecture is required using which scientists can automatically annotate sensor data through statistical modeling frameworks. Second, a scalable setup to support the processing of multiple data models concurrently. Third, seamless exchange of research results (model outputs and relevant metadata) among scientists. SensorFeed's architecture is designed to address these challenges.

Figure 1 presents an architectural overview of SensorFeed. It consists of the following three layers: *raw data acquisition and storage*; *model manager*, and *continuous query processor*.

Raw data acquisition and storage layer is responsible for capturing sensor readings from the acquisition hardware, convert the raw values to scientific measurements and finally store the results at each stage. The exact details of how this layer acquires raw readings are deployment dependent. Among common approaches, one can think of using TinyOS SerialForwarder (for TinyOS 1.x and 2.x compatible motes) to capture raw data directly from the motes. Other solutions normally involve using hardware specific proprietary APIs to read raw readings directly from remote sensors. Once the raw data is captured by SensorFeed, we need to calibrate the readings into measurements usable by scientists.

The calibration process is also hardware specific. Therefore, SensorFeed provides a calibration API using which users can easily implement custom calibration functions or reuse external data calibration packages. Because of this layer, raw data and its calibrated form with all the details, e.g. function parameters and constants, are stored in a persistent storage. Once a calibrated data stream is available, scientists can start building models to extract the meaning behind the values. Details of the data calibration process are depicted in Figure 2.

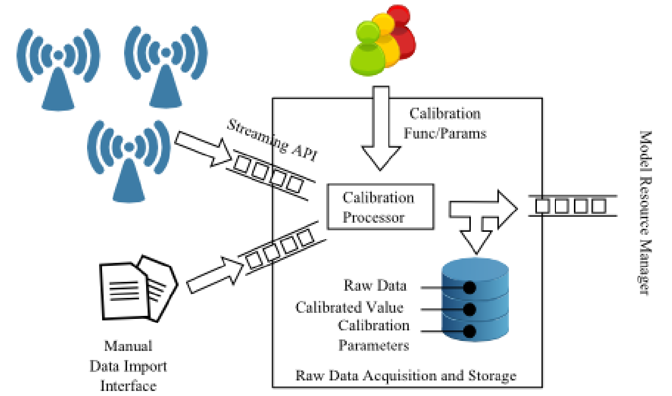

Fig. 2. Data calibration process

Model Manager receives streams of calibrated sensor readings on one side and statistical models on the other side. SensorFeed middleware is designed to concurrently process multiple sensor data models (Figure 3). To support the processing of multiple models within one middleware we rely on the existing works on managing multiple virtual execution environments for grid computing [17, 26]. Using a Virtual Machine (VM)-based approach, we can gracefully extend our setup as required by introducing new processing nodes into the grid. The models themselves have links to their relevant RDF repositories. The selection of repositories and RDF tags are done by each model independently.

To support this, SensorFeed provides non-blocking callback APIs for major statistical tools. These APIs allow scientists to receive streams of calibrated sensor readings as data arrives into SensorFeed. Once a model associates a RDF tag to a raw data that is just streamed into a modeling environment, scientists can use the non-blocking callback API to push the metadata into the system. The APIs are capable of buffering data streams in case the SensorFeed system does not reply or the call back connection is lost. Using persistent buffers to communicate between the middleware and the modeling environments ensures users from any potential loss of data. The non-blocking I/O also externalizes the data processing delay from the modeling environments.

Once SensorFeed receives data items with their tags from a modeling environment, it efficiently stores raw data into a persistent storage. More details of SensorFeed's storage layer and the associated data structure are provided in Section V.

SensorFeed is designed to support automated data sharing among scientists. To achieve this goal we rely on existing works on Data Stream Management Systems (DSMS), as shown in Figure 4. In a DSMS, local and remote users can express their interest through registering continuous queries. Compared to standard queries, continuous queries usually have two extra parameters—window-size, specifying the amount of data used at each processing stage and sliding-predicate, indicating how frequent a continuous query is to be evaluated by DSMS. Thanks to these two extensions, DSMS are usually more efficient and expressive compared to traditional database systems when dealing with continuous data streams. This is due to the fact that a DSMS requires a small snapshot of the recent history to produce results.

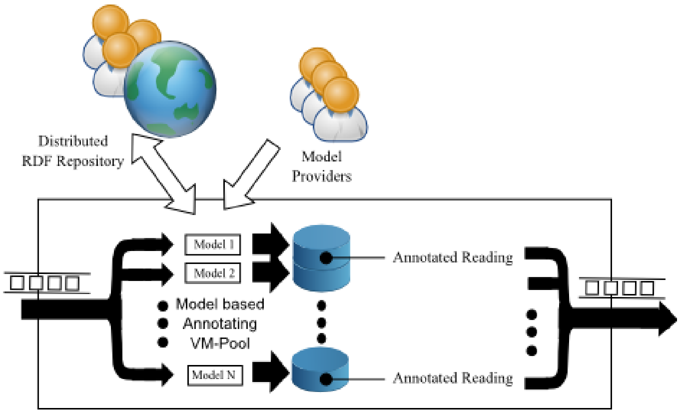

Fig. 3. Model-based annotation

Using SensorFeed, scientists can register queries which contain predicates involving conditions of interest. DSMS receives these queries and stores them into its query repository. When a new data stream is delivered to a DSMS, window manager is the first component which receives the data. Window manager then consults the query registry to build a list of potential queries which might be affected by the new data item. This list is then forwarded to query scheduler for further execution. After the execution, the actual results of each continuous query are streamed to the interested parties. To support these requirements, SensorFeed uses open-source data stream management system called GSN [2].

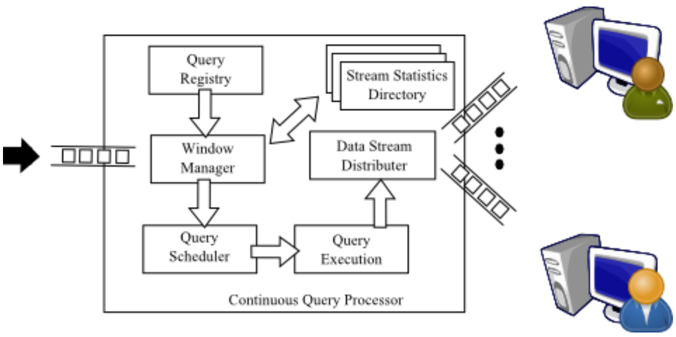

Fig. 4. Query processing

V. Raw and Metadata Storage Layer

To support efficient data storage in SensorFeed, we propose a specialized data structure. We call this storage layer (Figure 5) Multi-Stream Multi-Annotation (MSMA). It endeavors to reduce the replication of the metadata to a minimum, providing constant or logarithmic access time for popular operations in e-research applications. Its key operations are:

- Finding and browsing data items with certain annotations.
- Browsing among streams that share common annotations.
- Extracting values for several data streams at a particular timestamp.

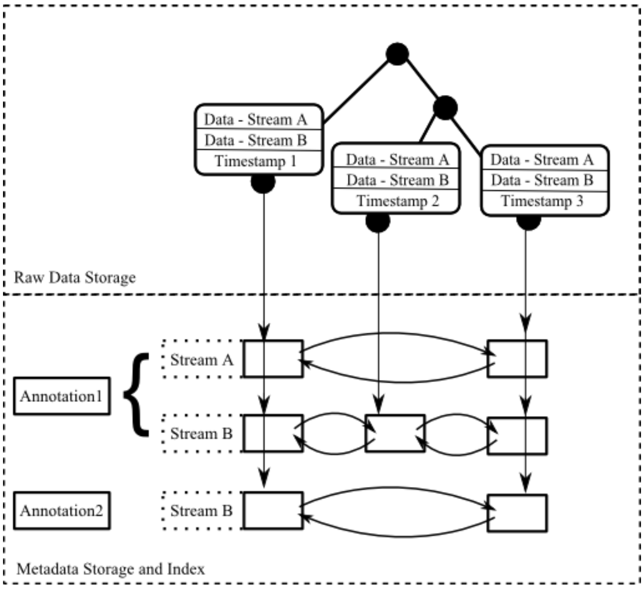

Fig. 5. Multi-stream multi-annotation storage layer.

To support these features we use a B+Tree structure to store raw data stream items for each timestamp at the leaves of the tree. A leaf of the B+Tree has a link to the metadata storage and index (MSI) component if any of the data streams in that leaf is to be annotated by a modeling environment. In this case there is a link to one of the annotations in the MSI. It is worth mentioning that, the order of the annotations is not important. All the annotations from all of the streams at a timestamp must be reachable by the relevant leaf of a B+Tree. To clarify this, let us consider the following example. In Figure 5, we have two data streams, Stream A and Stream B. Stream A and B both are annotated by Annotation 1 and Annotation 2 at timestamps 1 and 3. Furthermore, Stream B is annotated by Annotation 1 at timestamp 2 while Stream A is not annotated at this particular timestamp.

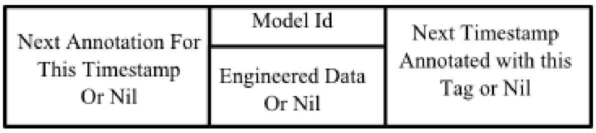

Fig. 6. Multi-stream multi-annotation storage layer.

To elaborate further Figure 6 presents a data cell at the MSI level. In order to make browsing large annotated datasets instantaneous for scientists, links to the next timestamp annotated with the same tag and other data streams that are annotated with the same tag at the same time instance are always available at each node at the MSI level. While these links introduce further storage requirements (4 bytes per link), they come with the added value that scientists can browse in datasets without having to wait for the system to find the next relevant data item. Browsing to the next data item with the same annotation would take constant time in MSMA thanks to the direct links in the cell definition as presented in the figure.

If an application cannot accommodate this extra storage requirements, one can stick to raw storage layer which is less preferment for browsing annotated sensor data. In this case locating the next data item from the current stream annotated with the same tag requires O(n) operations, n being the number of data stream elements. Locating other streams with the same annotation at a particular timestamp without using MSMA requires KLog(n) time, where K is the number of data streams and Log(n) is the look up time in the B+Tree.

VI. A Case Study

To be useful, the proposed architecture needs to account for how life scientists can use the sensed data. To this end, consider the CSIRO Phenonet project [13], which deploys a sensor network (Figure 7) over a field of experimental crops, monitoring plant growth and climate conditions.

The sensor network consists of sensors measuring solar radiation, air temperature, soil moisture, soil temperature, and an infrared sensor measuring leaf temperature. Based on the observed data, scientists are able to “map” microclimatic conditions such as light, temperature and soil moisture across the field to better evaluate and compare new plant varieties. By mapping these conditions and combining them with each plant’s genetic profile and performance, plant scientists can de-convolve the effects of microclimate and genome, thus improving the accuracy and speed of plant breeding. However, the collected raw sensor data is of little significance unless the data is processed to ensure meaningful representation.

Fig. 7. Sensor network deployment for the Phenonet project.

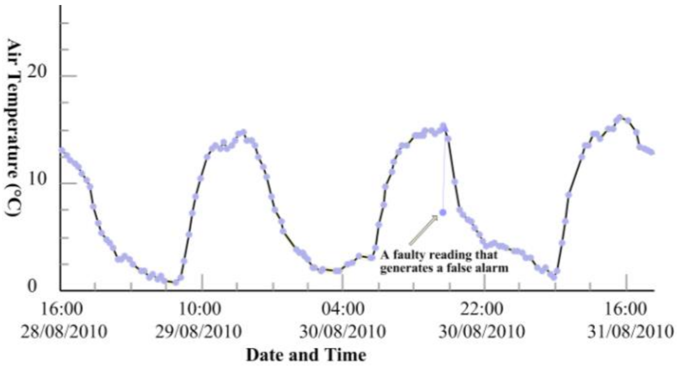

Fig. 8. Frost event detection with a faulty sensor reading.

In Phenonet, sensor data cleaning and annotating are done through use of RDF tags and R scripts. For example, severe frosts and heat events can damage the flowering mechanism of a plant, thus affecting seeds. Figure 8 shows air temperature observations at the Phenonet deployment in Leeton, NSW, plotted in 18 hours time intervals in 24-hour format to show frost scenarios, with temperature dipping towards 0°C.

Based on trends in observations (air, soil temperature and humidity measurements) R scripts can be used to associate data reliability tags to individual readings. The inputs of these scripts can come from multiple streams. The actual reliability estimation can be based on statistical models such as Exponential Moving Average (EMA)-based models.

A reliability estimating model associates a number to each individual sensor reading in the range of 0.0 to 1.0 as the measurement's reliability 1.0 being the most reliable reading and 0.0 being the least reliable reading. In the case of the Phenonet project in Australia, most frosts are surface radiative, where the ground temperature drops below the air temperature. Hence, a reliability tag is calculated through cross-correlating two data streams, air temperature and surface temperature.

Scientists use external R models to predict when frost or high temperature events may threaten crops. When such a frost event occurs, a warning in the form of an email or SMS can be sent to the researchers, prompting them to take necessary actions to save crops against adverse weather.

Using SensorFeed's automated model-based annotation, scientists can build data reliability aware models (models which work best on certain level of data). For instance, to avoid false alerts, they can employ sensor readings with reliability of 0.75 or higher hence avoiding faulty readings as highlighted in Figure 8. For daily lab-based analysis, scientists can decide to use readings with lower reliability.

It is worth emphasizing that thanks to SensorFeed's storage layer, external scientists can explore past sensor readings based on existing annotations, identify the sensor readings that have caused alerts in the past and re-evaluate sensor data to associate new annotations based on new data models. These new tags can be then reused by other scientists to further enrich the underlying sensor readings.

VII. Conclusion and Future Work

In this paper, we have presented a model-based sensor data enrichment middleware, called SensorFeed. Our work relates to sensor data management for e-research applications. SensorFeed is our first initiative for integrating metadata repositories and sensor data through the use of statistical modeling environments like Matlab or R. We are currently engaged in developing a prototype based on the proposed architecture. Together with plant scientists, we are aiming to deploy our final results in the Phenonet project. As future works, we will conduct an evaluation study of SensorFeed to better understand scalability and flexibility of our solution.

Acknowledgment

We would like to thank the members of the Phenonet project, in particular, Doug Palmer, Peter Lamb, Robert Furbank, Xavier Sirault, Leakha Henry, and Christopher Swain. This work is funded in part by National Collaborative Research Infrastructure Strategy (NCRIS) and the High Resolution Plant Phenomics Centre (HRPPC), Australia.

References

[1] D. Abadi, D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, "Aurora: a new model and architecture for data stream management," *The VLDB Journal*, vol. 12, no. 2, pp. 120-139, 2003.

[2] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," *Proc. International Conference on Mobile Data Management (MDM'07)*, 2007.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE communications magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[4] T. Anderson and H. Kanuka, *E-research: Methods, strategies, and issues*: Allyn and Bacon, Boston, USA, 2003.

[5] D. Arvind, A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, and J. Widom, "STREAM: The Stanford Stream Data Manager," *IEEE Data Engineering Bulletin*, vol. 26, 2003.

[6] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," *Proc. 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1-16, ACM Press, NY, USA, 2002.

[7] M. Botts, G. Percivall, C. Reed, and J. Davidson, "OGC® sensor web enablement: Overview and high level architecture," *GeoSensor Networks*, pp. 175-190, 2008.

[8] P. Buonadonna, D. Gay, J. Hellerstein, W. Hong, and S. Madden, "Task: Sensor network in a box," *Proc. 2nd European Conference on Wireless Sensor Networks*, pp. 133-144, 2005.

[9] S. Chandrasekaran, O. Cooper, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, F. Reiss, and M. Shah, "TelegraphCQ: Continuous dataflow processing," *Proc. International Conference on Innovative Data Systems Research (CIDR'03)*, 2003.

[10] N. Dawes, K. Kumar, S. Michel, K. Aberer, and M. Lehning, "Sensor metadata management and its application in collaborative environmental research," *Proc. 4th International Conference on e-Science*, pp. 143-150, IEEE CS Press, Los Alamitos, CA, USA, 2008.

[11] K. Delin and S. Jackson, "The sensor web: A new instrument concept," *Proc. SPIE Symposium on Integrated Optics*, 2001.

[12] M. Ellisman and S. Peltier, "Biomedical informatics research network," *Proc. 3rd International HealthGrid Conference*, 2005.

[13] B. Furbank, X. Sirault, and D. Deery, "Phenonet: A distributed sensor network for field crop phenotyping," *Proc. 1st International Plant Phenomics Symposium : from Gene to Form and Function*, 2009.

[14] L. Golab and M. Özsu, "Issues in data stream management," *ACM SIGMOD Record*, vol. 32, no. 2, pp. 5-14, 2003.

[15] R. Huebsch, B. Chun, J. Hellerstein, B. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. Yumerefendi, "The architecture of PIER: An internet-scale query processor," *Proc. 2nd Biennial Conference on Innovative Data System Research*, pp. 28-43, 2005.

[16] H. Jeung, S. Sarni, I. Paparrizos, S. Sathe, K. Aberer, N. Dawes, T. Papaioannou, M. Lehning, and E. de Lausanne, "Effective Metadata Management in Federated Sensor Networks," *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'10)*, 2010.

[17] I. Krsul, A. Ganguly, J. Zhang, J. Fortes, and R. Figueiredo, "Vmplants: Providing and managing virtual machine execution environments for grid computing," *Proc. IEEE/ACM Supercomputing Conference (SC'04)*, 2004.

[18] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the Kepler system," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1039-1065, 2006.

[19] S. Michel, A. Salehi, L. Luo, N. Dawes, K. Aberer, G. Barrenetxea, M. Bavay, A. Kansal, K. Kumar, and S. Nath, "Environmental Monitoring 2.0," *Proc. 25th International Conference on Data Engineering (ICDE'09)*, pp. 1507-1510, IEEE CS Press, Los Alamitos, CA, USA, 2009.

[20] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. Manku, C. Olston, J. Rosenstein, and R. Varma, "Query processing, resource management and approximation in a data stream management system," *Proc. International Conference on Innovative Data Systems Research (CIDR'03)*, 2003.

[21] NASA, "Directory interchange format (DIF) writer's guide," *Global Change Master Directory*, http://gcmd.nasa.gov/User/difguide/DIF_ Guide_2008.pdf, 2008.

[22] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, M. Pocock, A. Wipat, and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, vol. 20, no. 17, 2004.

[23] P. Pietzuch, J. Shneidman, M. Welsh, M. Seltzer, and M. Roussopoulos, "Path optimization in stream-based overlay networks," *Technical Report*, TR-26-04, Harvard University, 2004.

[24] E. Prud'Hommeaux and A. Seaborne, "SPARQL query language for RDF," *W3C working draft*, vol. 20, 2006.

[25] C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Sensor information networking architecture and applications," *IEEE Wireless Communications*, vol. 8, no. 4, pp. 52-59, 2001.

[26] A. Sundararaj and P. Dinda, "Towards virtual networks for virtual machine grid computing," vol. Proc. 3rd conference on Virtual Machine Research And Technology Symposium, 2004.

[27] M. Vardigan, P. Heus, and W. Thomas, "Data Documentation Initiative: Toward a Standard for the Social Sciences," *International Journal of Digital Curation*, vol. 3, no. 1, 2008.

[28] J. Wieczorek, M. Döring, R. D. Giovanni, T. Robertson, and D. Vieglais, "Simple darwin core," *Darwin Core Task Group*, http://rs.tdwg.org/dwc/terms/simple, 2009.

[29] A. Wombacher and P. Schneider, "Observation centric sensor data model," *Technical Report*, TR-CTIT-10-13, Centre for Telematics and Information Technology, University of Twente, The Netherlands, 2010.

[30] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292-2330, 2008.

[31] L. Q. Zhuang, J. B. Zhang, D. H. Zhang, and Y. Z. Zhao, "Data management for wireless sensor networks: Research issues and challenges," *Proc. International Conference on Control and Automation (ICCA'05)*, IEEE Press, NJ, USA, 2005.