

Collaborative Information Analysis for Sensor-Enabled Scientific Applications

Ali Salehi¹, Mukaddim Pathan¹, Dimitrios Georgakopoulos¹, David Deery²

¹Information Engineering Laboratory, CSIRO ICT Centre
108 North Road, Acton, ACT 2601, Australia

²CSIRO Plant Industry and High Resolution Plant Phenomics Centre
Clunies Ross Street, Black Mountain, ACT 2601, Australia

{ali.salehi, mukaddim.pathan, dimitrios.georgakopoulos, david.deery}@csiro.au

Abstract—Data collected from sensor networks are often analysed by cross-domain scientists who produce results that are requested by a variety of clients. In such a collaborative environment, scientific experiments include data collection from sensors, and data analysis performed by scientists. To meet the client requirements these activities have to be dynamically coordinated. Furthermore, this coordination must occur whenever data analysis results indicate that sensor data streams need to be adjusted to provide desirable results. In this paper, we present a platform and the design of its architecture that enable such real-time collaborative analysis of sensor data. We also discuss a case study from plant phenomics research. We illustrate that our solution permits scientists to build executable data models and conduct immediate data analysis that are driven by direct feedback from clients.

I. INTRODUCTION

The proliferation of diverse sensors and sensor networks [3, 32] has facilitated a paradigm shift in utilizing such technologies in research projects that involve scientists spanning interdisciplinary areas such as medicine, biology, chemistry, statistics, computing, electrical engineering and mechanical engineering. However, significant challenges remain before we realize the full potential of sensor networks in such cross-disciplinary research projects. For instance, many applications using sensor networks view a sensor network as a distributed data service [7, 33]. The data service abstraction for sensor network assumes that client applications submit their requests as queries and the sensor network responds with the requested data. This approach requires flexible mechanisms to permit service registration, discovery and advanced processing on the sensor network data [26, 28, 31].

E-research [4] applications often rely on the ability of scientists to collect, share, analyze, store and retrieve sensor network data. Increased levels of collaboration in performing these activities speed up the quantity and quality of e-research results. As e-research spans institutional, disciplinary and national boundaries, scientists in a wide variety of disciplines can benefit from new capabilities to interpret, correlate, and track heterogeneous sensor network data stored in digital repositories.

With the current data stream management systems for sensor networks, sensor data are made available to clients along with extensive data analysis. As a result of the data

processing and dissemination time associated with the analysis, scientists may have to use synthetic or similar data to perform their experiments in real time, often resulting in erroneous conclusions. Therefore, novel solutions are needed for immediate client review, feedback, and dissemination of sensor networks data in a dynamic and collaborative way to enable conducting experiments and produce results in real-time.

This paper presents our results in building a collaborative research platform for scientific experiments involving streaming data from sensor networks. Our approach aims at helping scientists in designing and deploying experiments that are governed by the requirements of their clients. We address the main challenges associated with the collaborative data analysis for sensor-enabled e-science applications, including:

- *requirements specification*—how clients capture and pass their requirements to scientists,
- *dynamic experiment creation*—how experiments are carried out based on the client-specified requirements, and
- *query processing*—how results are generated from data analysis of the conducted experiments.

We describe our contributions by:

- Presenting an architecture for model-based collaborative platform between scientists and clients (Section III).
- Describing methods for capturing experiment specification from clients and generating executable experiments through the use of scientific data modeling templates (Section II-A).
- Outlining a protocol for registering continuous queries and providing streaming results to data stream subscribers (Section III-C).

The rest of the paper is structured as follows. Section II describes the process of collaborative data stream analysis, along with an e-research case study. It is followed by the proposed architecture to assist collaborative data stream management. Section IV presents a comparative analysis of related works. Finally, the paper is concluded in Section V.

II. COLLABORATIVE DATA ANALYSIS

Effective collaboration between scientists and clients, including shared access to digital repositories storing analyzed sensor network data and the related resources is intrinsic to the

e-research vision. Collaboration may involve remote sensing, sharing of sensor network data, access to digital repositories, communication, and decision making for scientific experiments. The emergence of collaborative research in the sensor networks domain is driving the development of new data collections, metadata annotation giving meaning to data, and the standards and policies governing the storage, integrity, analysis, usage, access, discovery and the lifecycle management of data. Such collaboration makes use of distributed sensor networks, high performance computing infrastructure, scientific instruments and communications technologies to enable scientists to perform their research independent of time and geographic location.

Data management in a sensor network deals with the challenging task of defining how sensor-originated data are efficiently managed, stored and conveyed to the clients and scientists. In a collaborative project, it is particularly crucial to process and interpret sensor measurements for a scientific experiment. Activities related to managing sensor network data may be distributed in time and/or space [12]. Time distribution refers to activities taking place at different times, while being coordinated to have a unified effect. Space distribution means that activities may take place on different resources, while they are connected by a data network. 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, discovery, analysis and dissemination of research results.

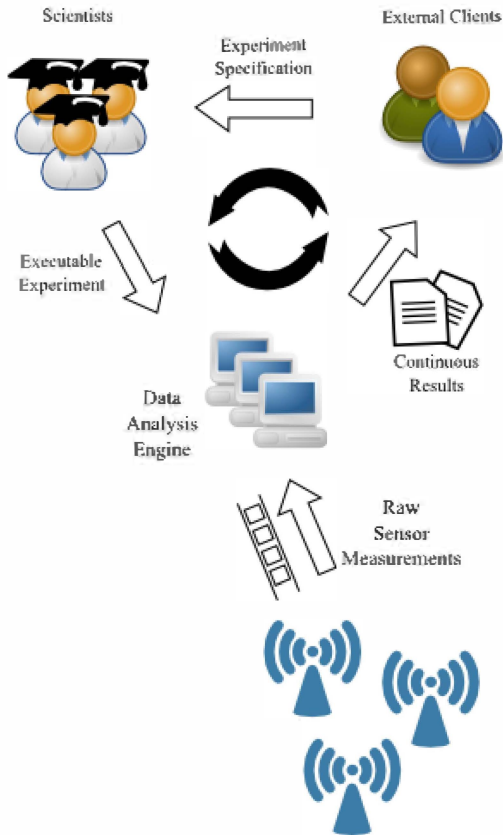


Fig. 1. The collaboration process

A. The Process

Figure 1 depicts how collaboration takes place among scientists and clients. It is an ongoing process that involves three steps stated in the following:

1. *Experiment specification:* E-research scientists receive the experiment specification from the clients. This specification contains client-specified data sources and model parameters, e.g. window size of a Fourier Transform-based model. It acts as input to the scientists to conduct their experiments.
2. *Experiment generation:* Upon receiving the client specifications, scientists build executable experiments based on model templates. This model templates provides the basic functionalities of a scientific experiment, but are customized according to the specifications from the clients (Figure 2).
3. *Result processing:* Scientists-built experiments are then executed and data analysis is performed on an analysis engine that holds the computational and processing ability to carry out a desired evaluation, based on raw sensor measurements. The experiment outcome is periodically streamed to the clients, who can generate continuous query to pick the required result. Clients can then give feedback to the scientists based on the obtained results and change their specifications, thus request the scientists to change the direction of an ongoing experiment or conduct further experiments.

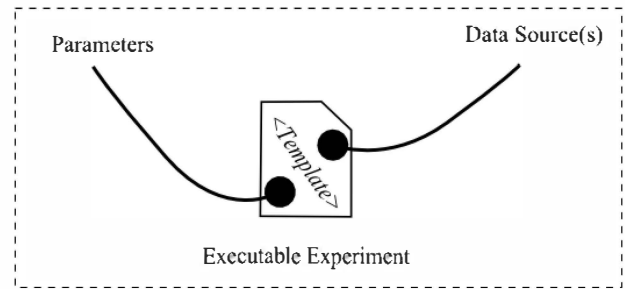


Fig. 2. Constituents of an executable experiment

The above collaboration process is a closed-loop control system that can save significant amount of work later in the data collection and processing sequence, as only the data of interest will be captured and analyzed. In our work, we seek to provide raw and processed sensor data, combined with collection reports and experiment details, to scientists, clients close to real-time or as soon as possible. This kind of simultaneous access to the partial-results of the experiments while the experiments are underway can give the opportunity to the clients to influence the direction of the experiments. Existence of sensor parameters and feedback channels help scientists in a collaborative project to understand the data on his/her own time without the communication delay.

B. An E-Research Case Study

Within the e-research community, a substantial amount of effort is spent in collaborative research projects. As a concrete example, we consider the Phenonet [14] project that deploys a

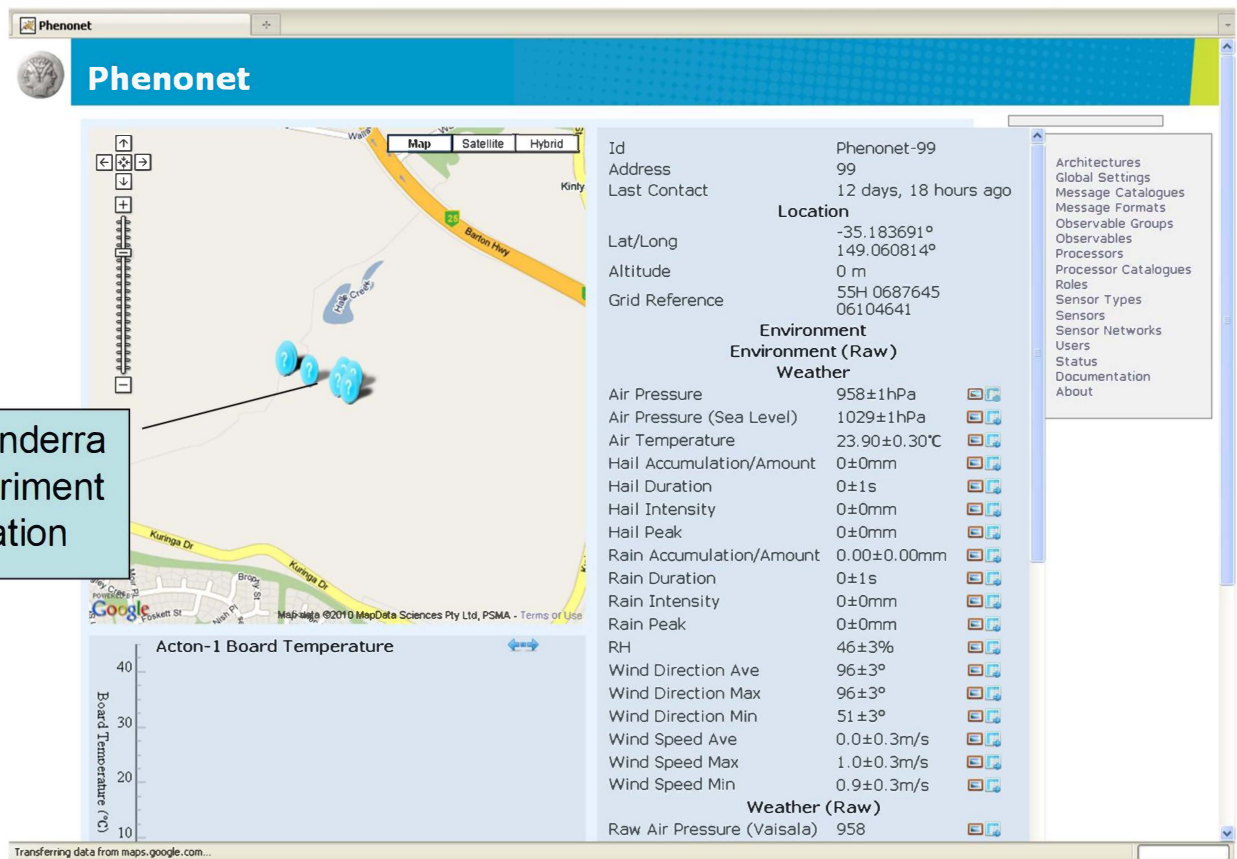


Fig. 3. Phenonet Graphical User Interface (GUI)

distributed sensor network (Figure 4) 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. The plant scientists involved in this project are interested in comprehensive sensor readings, based on a number of environment and water quality parameters, to run on yield and measure performance after frost, heat and drought. The observed micromet data is passed through various scientific models to analyze the heritability of traits in pre-breeding and breeding. The resulting data is relevant to a number of clients, such as plant biologists, environmental researchers, plant breeders, farmers, and funding bodies, who interact with the Phenonet deployment through a Graphical User Interface (GUI), as shown in Figure 3. Based on the observed data, e.g. soil moisture from different sensor deployment as shown in Figure 5, plant scientists are able to “map” microclimate 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.

A major challenge for this crop and field phenotyping system is the vast array of data types collected from a variety of sensors, both imaging, radiometric and genotypic

information. Therefore, the collected raw sensor network data is of little significance to cross-disciplinary scientists and clients unless the data is processed to ensure meaningful representation. As a result of detailed spatial and temporal analysis of environmental conditions, scientists provide different kinds of periodic reports concerning crop growth.



Fig. 4. Phenonet deployment

The reports are accessed by clients, i.e. non-scientists, in an ongoing basis. Clients can then provide feedback on the conducted scientific experiments to ensure that they receive the desired outcome related to crop performance. The feedback received from the clients is typically in one of the following forms:

- Continuing the experiment under a slightly modified environmental condition.
- Performing the data analysis with different environmental parameters.
- Conducting different type of data analysis on top of the raw sensor readings.

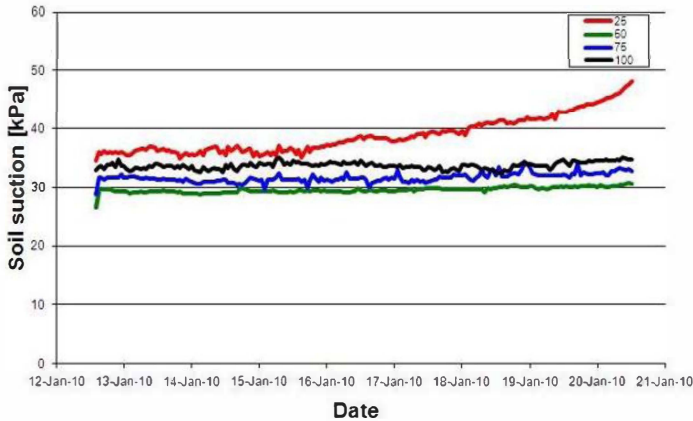


Fig. 5. Sensor observations for soil moisture

III. COLLABORATIVE DATA STREAM MANAGEMENT

To support the vision of having a middleware that enables scientists and clients to work together in building experiments and share results in real-time, we have to tackle the following three key challenges.

1. System architecture that that permits scientists to easily capture and calibrate sensor data.
2. Scalable setup to support the processing of multiple executable data models concurrently.
3. Seamless exchange of research results (model outputs and relevant metadata) among collaborators.

This section presents our proposed middleware architecture to address these challenges. While our focus in on sensor networks and streaming data, we endeavor to ensure that our approach can also be used by scientific experiments involving manual data measurements. We also focus on different perspectives of data analysis performed by the scientists.

Figure 6 depicts the architecture to assist collaborative data analysis for e-science applications. We base our proposed system upon existing sensor data management platforms for sensor networks. Our infrastructure consists of three distinct layers, namely, *raw data acquisition and storage layer*, *model execution manager* and *continuous query processor*. In the following sections, we present details of these layers and how they come into play to assist collaborative data stream management by the scientists in the Phenonet project.

A. Raw data acquisition and storage

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. In the case of wireless sensor networks, 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, it is passed through the calibration process which converts raw readings into measurements usable by scientists.

The calibration process is also hardware specific. Therefore, we provide a calibration API using which environmental scientists can easily implement custom calibration functions or reuse third party data calibration packages. At 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, it acts as the data source for model execution manager.

B. Model execution manager

Model execution manager receives streams of calibrated sensor readings on one side and executable experiments (as shown earlier in Figure 2) on the other side. In Phenonet, we are expecting to receive a number of experiment specifications from the clients, thus generating multiple executable experiments. We endeavor to provide an environment supporting the execution of those experiments within one middleware. For this purpose, we rely on the existing works on managing multiple virtual execution environments for grid computing [19, 29]. Using a Virtual Machine (VM)-based approach, we can gracefully extend our setup as required by introducing new processing nodes into the grid.

The actual communication between VMs and our infrastructure is done through the extensive use of non-blocking callback APIs. These APIs allow experiments to receive streams of calibrated sensor readings as data arrives into our system. As the actual execution of the experiment occurs inside a VM, a data connection is required to push the results of the experiment back to the system. To build such a data connection, we use persistent buffers to communicate between the middleware and VMs. A persistent buffer insures users from any potential loss of data in case the middleware is down. Moreover, the non-blocking buffers also externalize the data processing delay occurred in the VM from the overall infrastructure.

An experiment template can be applied to different data sources. An executable experiment contains the concrete details of the actual data sources used for each individual experiment template, along with any mandatory input parameters required by the experiment. Therefore, clients can launch several instances of the same experiment model using different data sources and parameters. To ensure that each instance of an execution template is isolated from the other instances, model execution manager instantiates a VM per experiment. Model execution manager is also responsible for locating the data sources used by the executable template and

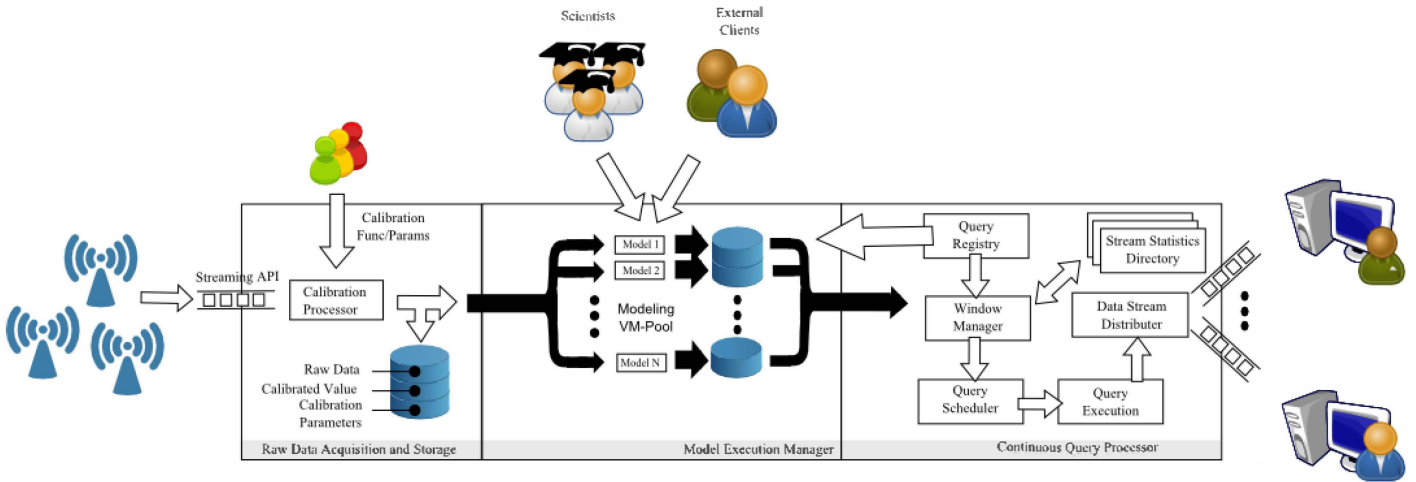


Fig. 6. The middleware architecture to assist collaborative data analysis

stream the data to the executable experiment. Once an executable experiment generates an output, it is buffered to generate continuous data streams consisting of the results. The output of each executable experiment is stored along with the experiment setup in a persistent storage.

C. Continuous query processor

In the Phenonet project, we need to support automated data sharing among scientists. To achieve this goal we rely on existing works on Data Stream Management Systems (DSMS). 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 can be used to generate continuous reports. For instance, clients can register a continuous query with a window-size and a sliding predicate of 7-days to periodically receive report concerning the results of an experiment every week.

As clients can register several queries concerning results of multiple executable experiments, we had to rely on a push based network protocol to continuously deliver the latest reports to them. Figure 7 presents our extension to an open-source data stream management system, called GSN [2], to support push-based data delivery of executable experiment results. As depicted, registration of a query consists of two stages. First, a stream consumer (client) posts a continuous query which has a window-size, sliding value and the name of the executable experiment. A stream producer (executable experiment generated by plant scientists) at the Phenonet project would response through an acknowledgement by generating a globally unique identifier (GUID) to the query and posting the GUID back to the stream consumer. The use of the GUID is important here as a stream consumer may ask for results of multiple executable experiments. The GUID is used by the stream consumer to match the results with the queries. A stream producer can reject a query registration request for several reasons such as insufficient permission

from the stream consumer or unavailability of the requested executable experiment, e.g. invalid query. To make sure the proposed middleware is not blocked by slow stream consumers, we seek to use UDP protocol to deliver the experiment reports. A continuous query has an indefinite lifetime. To support removal of the queries from the middleware, we rely on asynchronous communication between the data stream consumer and producer.

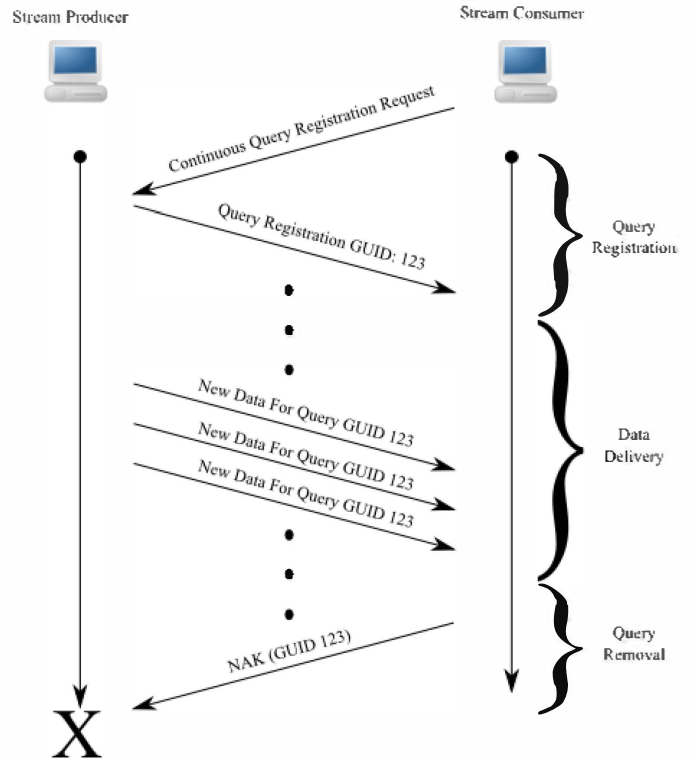


Fig. 7. Continuous query registration

Using the GUID accompanied with each data packet, stream consumer can check internally to see if the GUID is current valid, e.g. the query is not removed. If the GUID is current, the stream consumer continues processing the incoming data packet. If the GUID is expired not valid, e.g. the query is registered but later it is removed, a stream consumer responds with a NAK packet, indicating the stream

producer that a given data stream consumer is no longer interested in the query. Upon receiving the NAK packet, stream producer traverses its internal query repository to check if there is any other client interested in the results of the aforementioned query. If this is not the case, e.g. the last client interested to the query responded with a NAK, the query will be permanently removed from the stream producer. Such a query removal event can have further consequences in our system. For instance, if the models that are used to generate data for this query are no longer needed by any other query, the VM holding the executable experiments can get suspended indefinitely as long as they are not needed by any query.

On the continuous query processor side of the proposed middleware architecture, all the queries from clients are stored in a local query repository. When a data 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 clients and their queries which might be affected by the new data item. This list is then forwarded to the query scheduler for further execution. After the execution, the actual results, e.g. reports based on the experiment outcomes, of each continuous query are streamed to the interested parties.

IV. RELATED WORK

Data management is common in areas of distributed databases, distributed transaction processing and distributed file systems. Specifically, 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, 15, 22]. 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) [25] and Peer-to-peer Information Exchange and Retrieval (PIER) [16] 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.

The distributed and collaborative management of sensor network data in scientific and engineering applications have received substantial attention in recent past. Our work in the Phenonet project is comparable to initiatives such as soil moisture monitoring [17], integrated earth sensing [30], and sensor networks for agricultural system [24]. There also exist collaborative 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) [27] 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 of the above research works have mainly focused on off-line processing of sensor network data. However, sensor networks produce real-time data and this data is consumed in real-time by scientific experiments and client applications. For example: in a early warning system for frost events, sensor data processing, metadata annotation, feedback and result delivery have to be done in real-time. Based on the real-time processing of data, scientists can direct their experiments according to the ongoing events. In addition, 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, 18, 21], we seek to build our middleware 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.

There are also other research works in existing literature such as BIRN [13], Kepler [20] and Taverna [23] provide workflow-based collaborative data management infrastructure. 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. Although appealing, many of these research works are not replicable by scientists in collaborative e-research projects. While our work is based on plant research, we endeavor to build our middleware as a highly flexible, domain-independent collaborative data stream management system for sensor networks.

V. CONCLUSION

In this paper, we present a middleware architecture to assist collaborative data analysis between scientists and clients. We consider a real-world case study from agricultural engineering to demonstrate the applicability of our architectural approach. With our work, we intend to draw the research and development community closer to the operational community where interoperability, data links, bandwidth, reporting, decision making, and timing issues are of high significance. Along with the plant scientists, we are currently engaged in developing a prototype based on the proposed architecture. As future work, we will future explore the collaborative data stream management process in order to automate it with minimal human involvement. We will also integrate model-based metadata annotation of raw sensor readings during the collaboration process to make sensor network data and experiment results accessible and reusable even after a long period of time.

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, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393-422, 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] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the physical world," *IEEE Personal Communications*, vol. 7, no. 5, pp. 10-15, 2000.
- [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] P. Dourish, "The parting of the ways: Divergence, data management and collaborative work," *Proc. 4th conference on European Conference on Computer-Supported Cooperative Work*, p. 230, 1995.
- [13] M. Ellisman and S. Peltier, "Biomedical informatics research network," *Proc. 3rd International HealthGrid Conference*, 2005.
- [14] 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.
- [15] L. Golab and M. Özsu, "Issues in data stream management," *ACM SIGMOD Record*, vol. 32, no. 2, pp. 5-14, 2003.
- [16] 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.
- [17] B. Illston, J. Basara, C. Fiebrich, K. Crawford, E. Hunt, D. Fisher, R. Elliott, and K. Humes, "Mesoscale monitoring of soil moisture across a statewide network," *Journal of Atmospheric and Oceanic Technology*, vol. 25, pp. 167-182, 2008.
- [18] H. Jeung, S. Sami, 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] F. Pierce and T. Elliott, "Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington," *Computers and electronics in agriculture*, vol. 61, no. 1, pp. 32-43, 2008.
- [25] 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.
- [26] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," *Mobile networks and applications*, vol. 8, no. 4, pp. 427-442, 2003.
- [27] C. Shen, C. Srisathapornphat, and C. Jaikao, "Sensor information networking architecture and applications," *IEEE Wireless Communications*, vol. 8, no. 4, pp. 52-59, 2001.
- [28] J. Shneidman, P. Pietzuch, J. Ledlie, M. Roussopoulos, M. Seltzer, and M. Welsh, "Hourglass: An infrastructure for connecting sensor networks and applications," *Technical Report*, TR-21-04, Harvard University, 2004.
- [29] A. Sundararaj and P. Dinda, "Towards virtual networks for virtual machine grid computing," *Proc. 3rd conference on Virtual Machine Research And Technology Symposium*, 2004.
- [30] P. Teillet, R. Gauthier, and A. Chichagov, "Towards Integrated Earth Sensing: the Role of In Situ Sensing," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 1, pp. 249-254, 2002.
- [31] E. Wilde, "Making sensor data available using Web feeds," *Proc. 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'09)*, ACM Press, NY, USA, 2009.
- [32] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292-2330, 2008.
- [33] W. Yu, T. Le, J. Lee, and D. Xuan, "Effective query aggregation for data services in sensor networks," *Computer Communications*, vol. 29, no. 18, pp. 3733-3744, 2006.