ES3N: A Semantic Approach to Data Management in Sensor Networks

Micah Lewis[2], Delroy Cameron[2], Shaohua Xie[2], I. Budak Arpinar[1]

[1] LSDIS Lab
[2]Computer Science Department, University of Georgia,
Athens, GA 30602-7404
{lewis, cameron, shaohua, budak}@cs.uga.edu

Abstract. In this paper, we present a Data Management Tool called ES3N, which uses Semantic Web techniques to manage and query data collected from a mini-dome Sensor Network. Our tool supports complex queries on both continuous and archival data, by capturing important associations among data, collected and stored in a distributed dynamic ontology. The motivation behind our work stems from a desire to increase awareness of the advantages of Semantic Web techniques across the Sensor Networks spectrum and to highlight the inefficiencies in existing Data Management techniques in Sensor Networks for silos and mini-domes. We stress the advantages of semantics in this case study, and present a discussion that extrapolates the possible benefits Semantic Web techniques can bring to Sensor Networks in general.

Keywords: Sensor Networks, Semantic Techniques, Query Processing, Ontology, Semantic Association, OWL, RDF

1 Introduction

A sensor network is a computer network of many, spatially distributed devices using sensors to monitor conditions at different locations, such as temperature, sound, vibration, pressure, motion [15] etc. Two key issues in Data Management in Sensor Networks are Data Storage (*how to store data efficiently)* and Query Processing (*how to achieve fast and accurate information retrieval)*. The first issue is resolved with some efficiency, by storing data either locally or logically distributed at centralized locations. The second issue; Query Processing, is critical and central to our research.

Power conservation is always important to system performance in Sensor Networks. In Query Processing, the data manager is challenged to reduce and summarize data online while providing storage, logging, and auditing facilities for offline analysis [6] consuming minimal power. It must also provide an interface that allows a user to understand, collect, process and manage the status of the network and the data (such as averages, moments, histograms, or statistical summaries) generated on-the-fly in real time [6]. Most Query Processing languages are based on some form of SQL-like syntax. For example, work on the TinyDB Project [10] at UC Berkeley and The Cougar Project [5] at Cornell University outline a query language that

consists of **SELECT-FROM-WHERE-GROUPBY-HAVING** blocks to support selection, join, projection, aggregation, and grouping [6]. This language is efficient for database oriented storage mechanisms. If the user poses a query such as SELECT date, time FROM database WHERE date = "11-20-05" AND temp = 60, the query will return the expected results, but using mere string and integer comparisons. This approach ignores any relationships between the two pieces of data. For example, the date may be related to temperature by a *has_temp* relationship allowing node to node connections via edges. In this paper, we identify and exploit such relationships by searching data semantically.

We focus our case study on The National Peanut Research Laboratory (NPRL[1]) in Dawson, Georgia, which uses a Sensor Network across a mini-dome to monitor conditions affecting peanuts. Our paper is significant for the following reasons:

- We develop an alternative storage mechanism in the form of an ontology, for storing data
- We illustrate semantic query processing by exploiting semantic associations between data
- We encourage the audience to consider semantic techniques in resolving the larger data management and query processing issues affecting Sensor Networks

2 Motivation

This paper exemplifies how the use of semantics can enhance data management in sensor networks. Semantics exploit underlying relationships between data captured by sensors, creating a versatile framework that can be utilized in various applications. We devote our attention to grain and seed storage, and show how our system approaches data storage and data management in this application.

2.1 Absence of Data Storage

The initial motivation for this research was inspired by interaction with Cargill[2], an international provider of food, agricultural and risk management products, and services [4]. This corporation stores cereal grain and oil seed products in large storage silos, and their goal is to ensure that the stored products are kept at premium quality before distribution. Guidelines in storage conditions are enforced by the Grain Inspection, Packer and Stockyards Administration (GIPSA) and the American Society of Agricultural Engineers (ASAE). These guidelines include equilibrium moisture content and upper bounds on temperature and relative humidity.

---

[1] *NPRL was established in 1965, and current research centers on detection of mycotoxin and aflatoxin in peanuts*

[2] *Cargill is involved in every step of the production process, from harvesting to distribution - http://www.cargill.com/*

Cargill uses rather primitive data acquisition methods. Data are retrieved by random sampling via hand held sensors, which often yield an inaccurate representation of conditions. This makes it difficult to respond to deteriorating conditions in their early stages. Also, there are no records of historical data to aid in future decision making. We propose solutions to both these problems by utilizing a distributed Ontology as a storage repository that uses semantics to discover relationships among the streaming data.

2.2 Data Management Inefficiency

Our awareness of some of the problems affecting data management in Sensor Networks was amplified after communication with the USDA Agricultural Research Service (ARS) NPRL, which also engages in sub-par data management practices. Readings taken from sensors hourly are analyzed to determine required action, and historical data are available for analysis as well. The drawback is that Microsoft Excel Spreadsheets are used as the mechanism of storage; and they are not user-friendly for querying historical data. Therefore, with the use of a distributed Ontology, we resolve these issues by employing a Semantic Search technique allowing the user to easily query historical data. Through this system, simple and complex range queries are supported. Figure 1 illustrates the configuration of the NPRL mini-dome Sensor Network.

3 Overview

The development of ES3N followed a multi-layered process involving the following steps:

1. *Data Collection* - usually a significant issue in Sensor Networks. Resolving Heterogeneous data by tagging is often laborious. Raw data collected from NPRL, as text and Excel files assumed to be accurate, eliminated the need to focus on data collection challenges affecting this Sensor Network.

2. *Memory Caching* - efficient query processing requires efficient use of main memory. Real-time streaming warrants some data in main memory, but simultaneous permanent storage is necessary for efficient memory usage.

3. *Data Tagging* - in a small-scale sensor network, it may be reasonable to have homogeneous sensors which are usually identical or similar in terms of function [6]. In reality, Sensor Networks are inherently heterogeneous; fragments of data from particular nodes within the network must be given unique id’s for proper identification.

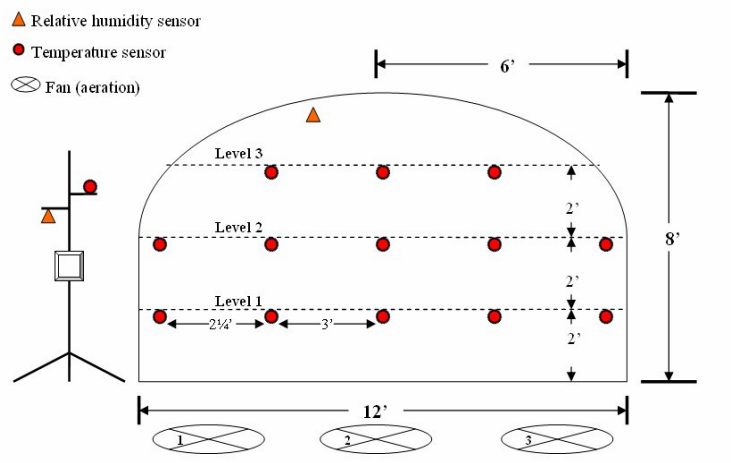

Fig. 1. Mini-Dome Sensor Layout Schematic

4. *Ontology Representation* - depending on the application, it can be necessary to import or export data using standards such as RDF/RDFS and OWL [1]. For our purposes both OWL and RDF were necessary.

5. *Query Processing* - this continues to be a central issue for researchers in computer science and information systems. Query optimization is central to all areas of computing. We use Semantic Techniques in this approach.

6. *User Interaction and Data Representation* - a wide range of visualization tools are available. Choosing among graphs, touch graphs, web interfaces, histograms, and tables will depend on the information being conveyed. We develop an interface we call ES3N[3], to present data to the user.

7. *Evaluation* – Analysis and evaluation of Semantic applications are void of existing benchmarks and measurement standards. Often times, the best measure of performance and efficiency are attained through human subjects. In this instance we use empirical results to evaluate ES3N and present some discussion that augments these results. Figure 2 outlines our multi-layered approach.

3.1 Querying Sensor Networks

Sensor Networks are typically able to process a vast range of queries fairly efficiently using existing techniques. The TinyDB project is based on a query language that

[3] *The acronym ES3N was kept from a previous version of our paper entitled Exploiting Semantics in Sensor Networks for Silos.*

supports basic, aggregate, temporal aggregate, event based and even lifetime-based querying capabilities. This language supports a range of query types, including monitoring, network health, exploratory, actuation and offline delivery queries. From a Semantic Web perspective, these querying types can be further leveraged by discovering and using Semantic Associations between fragments of data, to present greater query richness. We demonstrate this in our layered approach.

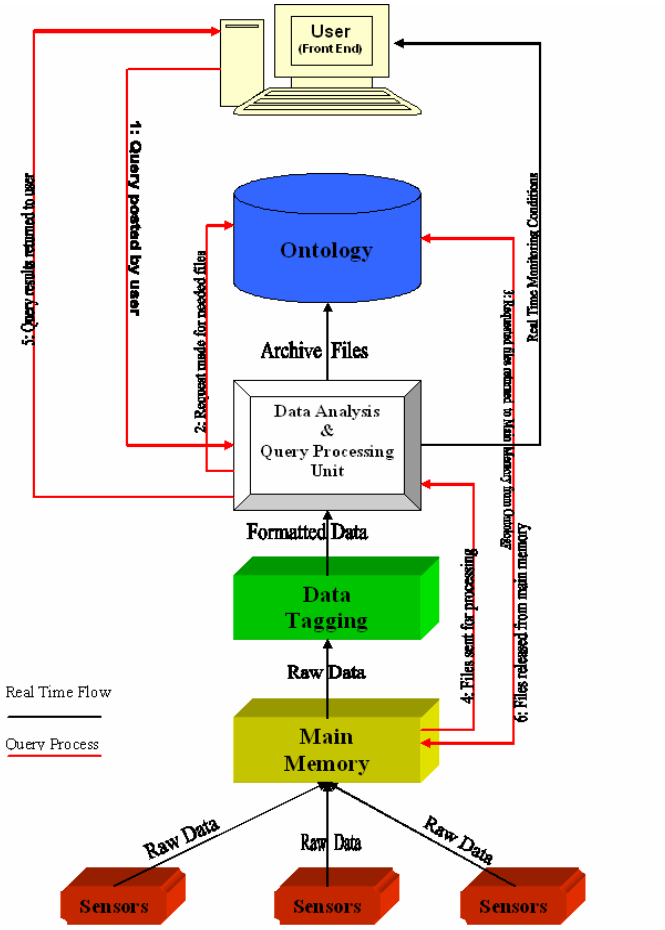

Fig. 2. Multi-layered approach to application development

4 Implementation

4.1 Data Collection

ES3N collects data by storing them in an ontology. Daily RDF files are written to a special data repository, and only imported into main memory depending on the nature of the posted query. RDF files in main memory are quickly released back to permanent storage after use, importing the next file required.

4.2 Main Memory Caching

Streaming data come directly into main memory, where it is routed to the user interface and duplicated in the ontology. Data must be archived for two reasons; first, to support historical queries and more importantly, to manage memory efficiently. Archiving in the form of daily RDF files makes it unnecessary to keep yesterday's data in memory. Streaming data is temporarily in main memory while archival data migrate to disk. This is also an efficient way of indexing incoming data.

4.3 Data Tagging

In most Sensor Networks raw data are heterogeneous and can pose problems for data identification. Fortunately, NPRL uses two distinct types of node sensors; thermocouple sensors for temperature and RH sensors for relative humidity. Incoming data are therefore assumed to be accurate. However, incoming data are time stamped so that the *has_date* and *has_time* relationships refer to unique literals for each entity, and all other edges connect non-distinct literals.

4.4 Ontology Representation

An Ontology is a formal explicit description of concepts in a domain of discourse [13] and remains a primary contributor to the development of in Semantic Web Applications. According to Natalya F. Noy and Deborah L. McGuiness[13], the development of ontologies is important for sharing common understanding of the structure of information among people and software agents, enabling reuse of domain information, making explicit domain assumptions and analyzing domain knowledge[13]. For our purposes the most important contribution of an ontology is its ability to hold entities with relationships and constraints among them. For example, Table 1 shows constraints obtained through empirical research at NPRL on peanut storage. Using the *has_max_temp* and *has_max_humidity* relationships in the ontology these condition constraints play a key role in Actuation[4] queries.

[4] *An Actuation Query requires a physical action depending on query results*

Table 1. Peanut Constraints

Property	Maximum Value
Relative Humidity	80%
Temperature	75

To implement our application, we use Protégé 3.2[5] which is a free, open source ontology editor and knowledge-base framework developed and distributed primarily by Stanford University. We choose Protégé 3.2 because it allows the creation of an ontology schema, which can be exported easily to OWL and RDF/RDFS formats. Our ontology schema consists of a set of predefined classes, attributes and constraints exported as an OWL file. Incoming data from the NPRL sensor network are compared against this schema, before being written to disk. A single record is described as shown in Figure 3.

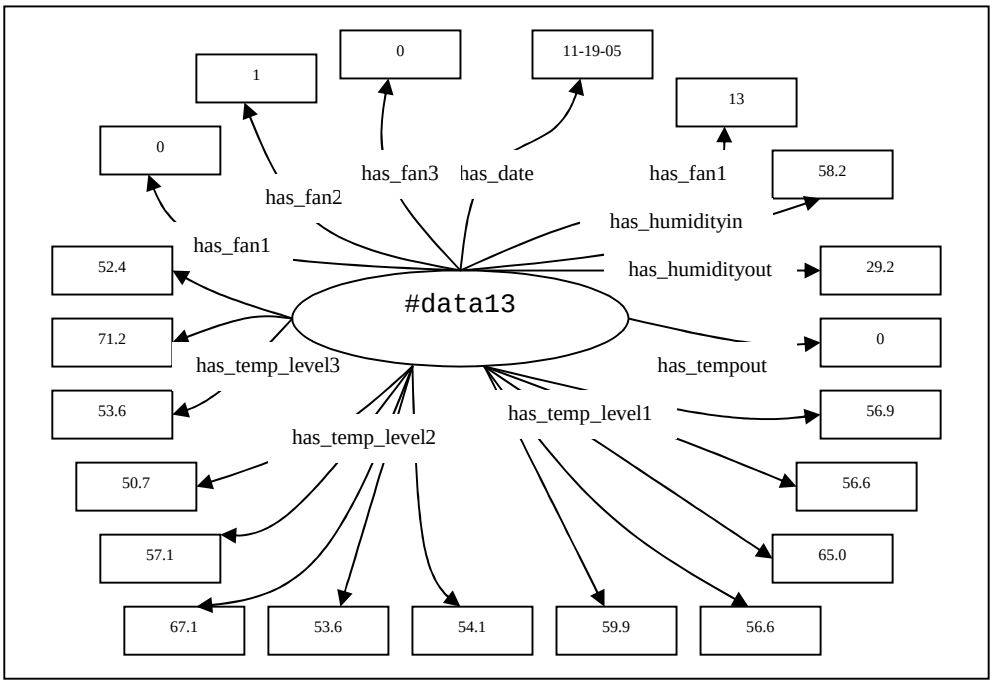

Fig. 3. Single Record in ES3N Ontology

[5] *Protégé 3.2 is a beta version of Protégé from Stanford University*

4.5 Query Processing and Data Analysis

For Query Processing, we use SPARQL, which is a Protocol and Query Language designed for accessing RDF data. In fact, SPARQL is a Semantic Web candidate recommendation presently undergoing standardization by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium [15]. SPARQL is embedded in Jena, which is a Java framework for building Semantic Web applications that provides a programmatic environment for RDF, RDFS and OWL, including a rule-based inference engine [8]. Our Ontology Schema, once imported in main memory, creates an ontology model[6], to which formatted streaming data are added as *resources*. SPARQL queries capture relationships among data triples. Consider four RDF resources in Figure 4.

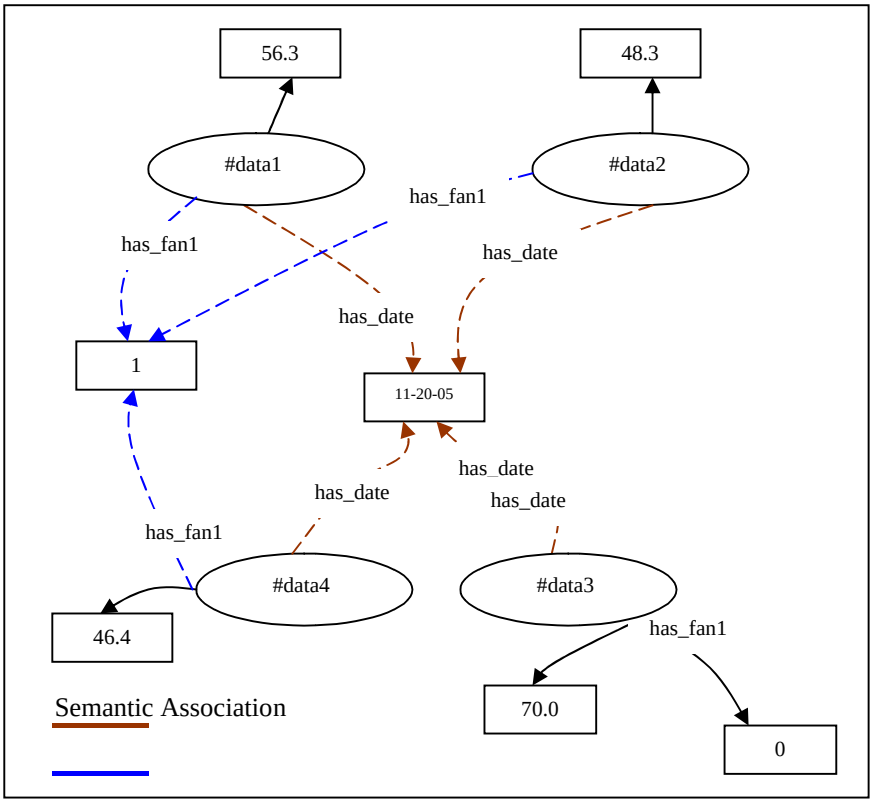

Fig. 4. Semantic Association among four nodes

The four resources, *data1*, *data2*, *data3* and *data4* share all of the same properties. In particular, the literal value (11-20-05) for the *has_date* relationship is the same for

[6] *An ontology model is a data container that holds resources about the ontology.*

all entities. A query posted to the model based on this date, would form an association among the four entities because their edges meet at the same node.

This kind of semantic association points to the possibility for supporting much more complex queries that can traverse several hops, following many nodes and edges to a variety of end points. SQL-like query languages, like those suggested in the TinyDB and Cougar project, are limited merely to string and integer comparisons, and are therefore incomparable to semantic search.

4.5.2 Exploratory Query

The simplest form of query retrieves a single record from the ontology. Consider the following query: Query = "return the record on 11-22-05 at 5:00am."

In answering this query, ES3N imports the archive file 11-22-05.rdf into the main memory model. SPARQL extracts the appropriate record based on the *has_date* and *has_time* properties.

4.5.3 Monitoring Query

A more complex query searches and returns more than one record within the same RDF file. Consider: Query = "return all records for 11-22-05."

SPARQL traverses the entire 11-22-05.rdf file and returns all the records in that file to satisfy this query.

4.5.4 Range Query

A typical range query requires importing several RDF files into the model and returning the data satisfying the query in each file. For example: Query = "return all days in the month of December when only fan2 was on." ES3N performs these steps:

- Import first file, 12-01-05.rdf
- Query this file for results
- Release this file back to disk
- Repeat the previous steps on the next file, 12-02-05.rdf and all remaining December files

Thus far, we have shown it is possible to pose three different types of queries using ES3N. We present these results to the user interface for analysis.

4.6 User Interaction and Data Representation

The ***ES3N Interface*** is a user-friendly front-end that allows data viewing in real time, as well as querying and viewing of historical data. This GUI serves as a forefront, allowing the user to utilize the research presented in this paper.

The real-time monitoring system is a preventative system versus a reactive one. The streaming data can be analyzed to determine corrective action, primary among which is Aeration. Aeration is used to maintain an equilibrium between the temperature inside and outside the mini-dome, reducing the chances for an increase in moisture of the stored product and buildup of condensation [2]. Three fans located under the perforated floor of the mini-dome pull air down through the stored product, when needed. Thus, the mini-dome is divided into three sections (left, right and center) which are each aerated by one of three fans. The calculation of the average temperature of the mini-dome as a whole is calculated as such,

$$avgTemp = \frac{\left(\sum_{i=1}^{2}\sum_{j=1}^{5} Level_i Sensor_j\right) + \sum_{j=1}^{3} Level_3 Sensor_j}{\#\text{ of thermocouple sensors}}. \quad (1)$$

Color-coded flags show the overall status of the stored product. Yellow signifies that the average temperature of the mini-dome as a whole is less than the outside temperature, but one section has a higher average temperature than the outside temperature. In this case the corresponding fan is turned on. Orange signifies that two sections are observed to have a higher average temperature than the outside temperature; and in this case two fans are activated. Finally, red signifies that the average temperature of the mini-dome as a whole is greater than the outside temperature. In this scenario, all three fans are turned on to lower the temperature. However, if the relative humidity of the air outside is greater than the maximum relative humidity allowed for the stored product, no aeration takes place. This is due to the fact that with a relative humidity over 80%, the air is very moist [2]. Therefore, aeration would bring in moist air, causing more harm than help. Figure 5, shows a snapshot of the ES3N Interface.

The ES3N Interface also allows the user to submit queries to the system. The user may query historical data in the form of range or monitoring queries. Finally, the ES3N Interface shows an overview of the ontology file system, which stores data in two formats, text and RDF. The RDF format is merely an RDF file containing the instances of that day. The text format shows the instances in tuple format, just as the query results on the query page.

5 Discussion

A Semantic approach to the Data Management issues facing the NPRL Sensor Network, leads us to highlight the following important aspects of the Semantic Web.

Ontology Data Storage - With a growing number of software engineers and designers continuing to embrace the Semantic Web, ontology data storage is the wave of the future. Oracle recently announced incorporating ontology building tools in its database management software. Protégé continues to build on earlier versions geared toward providing user friendly ontology editors. A growing awareness through conferences, workshops and exchange of information continues to champion the cause of the Semantic Web. Ontology schemas, OWL and RDF ontologies offer an alternative storage mechanism, annotating data and taking advantage of existing relationships among them. These provide immeasurable benefits for a new form of query processing, as demonstrated in our work.

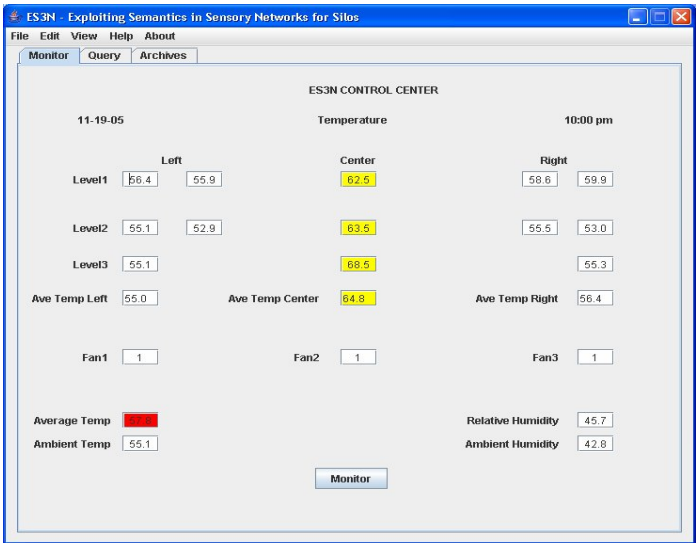

Fig. 5. ES3N screenshot

Semantic Query Processing – SPARQL (www.w3.org/TR/rdf-sparql-query/) and other RDF query languages, including RQL [9], RDQL [14], and SquishQL [11] continue gaining greater recognition and popularity in many Java applications such as Jena and Sesame. Publicly available Semantic Query Processing tools hiding technical design from the user will unmask the future of query processing, increasing query richness immensely.

The future of Semantic Web - Semantic Association continues to be a major concept used in a wide variety of applications. Current research at the University of Georgia encompassing Conflict of Interest Detection [1], National Security [16], Insider threat and Semantic Discovery (SemDis) are pivotal among these. The Key issues such as standardization through Semantic Annotation and Entity Disambiguation remain critical to the future of the Semantic Web. Accessibility and availability of larger standardized ontologies will enable searching multiple data repositories in many different formats. In our research we have shown that Sensor Networks too can benefit from Semantic techniques, and we hope to stimulate thought from the experts in this field.

6 Conclusions and Future Work

Our ES3N application only demonstrates its data management capabilities with homogeneous sensors. In more complex Networks, sensors are undoubtedly heterogeneous. Our ontology schema allows for application across a multi-grain platform bringing the issue of ontology size into question. Years of collected data allow for much larger range queries requiring faster response times and dealing with larger data files. Future works concern using larger OWL and RDF-based storage mechanisms to support a multi-grain platform and handling much larger RDF files. In particular, we consider using BRAHMS [7], which is an application developed by research at the University of Georgia to store extremely large RDF data. BRAHMS boasts size and speed, and is gaining ground as the recommended tool for RDF storage.

In conclusion, we have shown that Semantic Techniques are important in forming associations among data adding to the richness of the querying capability. Our distributed ontology philosophy offers fast and efficient data retrieval, limiting the use of main memory in answering complex queries. Our future work promises to address larger Ontologies and investigating more complex Semantic Associations.

8 References

1. Aleman-Meza, B. et al.: Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. ACM (2005)
2. Arthur, S.L., Brown S.L., Butts, C.L., Dorner J.W.: Aerating Farmer Stock Peanut Storage in the Southeastern U.S. (2006) *Trans. ASABE* Vol 49(2)
3. ASAE Standards 2002: 49th Edition, Standards Engineering Practices Data. American Society of Agricultural Engineers
4. Cargill Industries Inc. Agricultural Commodity Trading. cargill.com
5. Demers, Alan.et al.: The Cougar Project: A Work-In-Progress Report. ACM SIGMOD Record, 32(4): 53-59 (Dec 2003)
6. Gehrke, Johannes and Madden, Samuel.: Query Processing in Sensor Networks. IEEE CS and ComSoc (2004)
7. Janik, M. and Kochut, K. BRAHMS: A WorkBench RDF Store and High Performance Memory System for Semantic Association Discovery. Fourth International Semantic Web Conference , Galway, Ireland (2005)
8. Jena – A Semantic Web Framework for Java. jena.sourceforge.net
9. Karvounarakis, G. Alexaki, S., Christophides, V., Plexousakis, D. and Scholl, M.: RQL: A Declarative Query Language for RDF. Eleventh International World Wide Web Conference, (Honolulu, Hawaii, USA, 2002), ACM
10. Madden, Samuel R. et al.: TinyDB : An Acquisitional Query Processing System for Sensor Networks. ACM Transactions on Database Systems, (2004)
11. Miller, L., Seaborne, A. and Reggiori, A.: Three Implementations of SquishQL, a Simple RDF Query Language. First International Semantic Web Conference on The Semantic Web, (Sardinia, Italy, 2002), Springer-Verlag, 423 – 435.
12. Ni, Lionel M. et al.: Semantic Sensor Net : An Extensible Framework. IEEE International Conference on Computer Networks and Mobile Computing, Zhangjiajie, China (Aug 2005)
13. Noy, Natalya and McGuinness, Deborah L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford, CA

14.Seaborne, A. RDQL – A Query Language for RDF, 2004
15.“Sensor network”. Wikipedia, the free encyclopedia. en.wikipedia.org (2006)
16.Sheth, A.P et al.: Semantic Association Identification and Knowledge Discovery for National Security Applications. Journal of Database Management, 16(1). 33-53