EFFICIENT MONITORING OF SEED AND GRAIN STORAGE FOR THE PREVENTION OF PRODUCT DEGRADATION

by

MICAH ANTHONY LEWIS

(Under the Direction of Walter D. Potter)

ABSTRACT

Various oil seed and cereal grain products are stored by farmers and commercial manufacturers each year, world wide. However, it is while in storage that these products are most susceptible to quality degradation or even spoilage. Present detection methods, such as random sampling, are not practical for preventing such events during storage. Therefore, our system was designed to rectify this occurrence. The system is a conglomerate of real-time monitoring and data management. Utilizing a sensor network, streaming data are analyzed to determine the conditions of the stored product and decide if any corrective action is needed, such as aeration. The system also affords the functionality of storing and querying historical data. This thesis explains how various computer science concepts were pooled together to construct this real-life application.

INDEX WORDS: Sensor network, USDA, Real-time monitoring, Semantic web, RDF, Cereal grain, Oil seed, Grain spoilage, Ontology, OWL

EFFICIENT MONITORING OF SEED AND GRAIN STORAGE FOR THE PREVENTION OF PRODUCT DEGRADATION

by

MICAH ANTHONY LEWIS

B.S., Albany State University, 2004

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2006

EFFICIENT MONITORING OF SEED AND GRAIN STORAGE FOR THE PREVENTION OF PRODUCT DEGRADATION

by

MICAH ANTHONY LEWIS

Major Professor: Walter D. Potter

Committee: Budak Arpinar
Samir Trabelsi

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2006

DEDICATION

This paper is dedicated to my lovely fiancée, LaToya N. Henderson, and the Lewis family of which I am the youngest child.

Mr. & Mrs. Elijah P. Lewis (Dad & Mom)

Mr. & Mrs. Jeffery Grace and Corinthian
(oldest & only sister, husband and daughter)

Mr. & Mrs. Elijah L. Lewis, Mary Joy and Bonita
(oldest brother, wife, and daughters)

Mr. Lendon L. Lewis (brother, middle of all children)

Mr. & Mrs. Frederick Lewis and Xavier
(youngest older brother, wife and son)

To all of you, I thank you for being my backbone throughout the years.

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge God for giving me the gift of life and showing me His favor throughout my entire life. Without Him, I could surely do nothing.

I would like to thank Ms. Gwendolyn Campbell for her encouragement and advisement throughout my undergraduate studies at Albany State University. I would like to thank Dr. Potter for believing in me even since CSCI 6530, a course taken my first semester in graduate school before he became my major advisor. He has an uncanny wit that knows no boundaries. I am honored that he has trusted me with his ideas and advice. I also want to thank my other committee members, Dr. Budak Arpinar and Dr. Samir Trabelsi. Dr. Arpinar, thank you for your encouragement to pursue my first publication and for your motivation in CSCI 6370 and 8380. Dr. Trabelsi, thank you for the privilege to work beside you daily and learn countless things about agriculture, engineering, microwaves, Europe, and life.

Lastly, I would like to acknowledge the two co-authors of my first publication, Delroy Cameron and Shaohua Xie. Shaohua, thanks for your never ending enthusiasm to get things done. You kept us on schedule and came through with major advancements when we needed them. Delroy, thanks for your tireless nights of hard work throughout the spring and summer; but most of all, thank you for your friendship.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER

1 INTRODUCTION

1.1 Problem at Hand 1

1.2 Background and Problem Domain 2

1.3 Existing Techniques 6

1.4 Related Work 8

2 SOLUTION APPROACH 17

2.1 System Design 18

3 SENSOR NETWORKS 22

3.1 Sensor Network Background 22

3.2 Uses of Sensor Networks 24

3.3 Sensor Nets in Real-time 26

3.4 Sensor Network: Mechanism for Data Acquisition 27

4 DATA STORAGE AND MANAGEMENT 33

4.1 Evolution of the Semantic Web 33

4.2 Descriptive Languages: RDF, RDFS and OWL 38

4.3 Ontologies 42

4.4 Data Storage, Semantically 44

4.5 Meaningful Data Management...51

5 RESULTS AND EVALUATION..56

6 CONCLUSION AND FUTURE ENHANCEMENTS...64

BIBLIOGRAPHY...67

APPENDIX

A ONTOLOGY SCHEMA FOR DATA INSTANCES...71

LIST OF TABLES

Page

Table 5.1: Query Response Evaluation Table..61

LIST OF FIGURES

Page

Figure 1.1: Fungal Growth within a Corn Sample....4
Figure 1.2: Unloading Reveals Molded Mound....7
Figure 1.3: SiloTrack Main Display....9
Figure 1.4: Silo Details Frame....10
Figure 1.5: GrainPlan Main User Interface Page....12
Figure 1.6: Individual Grain Store Page....13
Figure 1.7: Photograph of Infrared Beam Cross-Section....15
Figure 2.1: Complete System Diagram and Data Flow....19
Figure 3.1: Mini-dome Schematic and Sensor Layout....28
Figure 3.2: Monitor Page of GUI....30
Figure 3.3: Active Monitor Page....32
Figure 4.1: RDF Graph Representing Eric Miller....39
Figure 4.2: Class Hierarchy....43
Figure 4.3: View of Archive Files within Ontology....49
Figure 4.4: Illustration of Data Resources and Properties....50
Figure 4.5: File Routing During Querying....52
Figure 4.6: GUI Query Page....53
Figure 4.7: Semantic Association Aids Querying....54
Figure 5.1: November 19, 2005 Fan Correlation....57

Figure 5.2: December 8, 2005 Fan Correlation 58

Figure 5.3: Query Efficiency Evaluation...60

Figure 5.4: Excel Snapshot of Data from 11/19/05 62

Figure 5.5: Snapshot of Query of Data from 11/19/05 63

CHAPTER 1

INTRODUCTION

1.1 Problem at Hand

The objective of this research was to design a system to address the problems currently faced in the storage of oil seed and cereal grain products. The problem was first introduced to us through communication with Cargill (Cargill, 2006), an international provider of food, agricultural and risk management products and services, and the USDA Agricultural Research Service (ARS) National Peanut Research Laboratory (NPRL) in Dawson, GA. Both facilities are involved in product storage: Cargill works with many types of seed and grain but mainly soybean, wheat and corn, and USDA works mainly with shelled and unshelled peanuts. However, Cargill's storage is for commercial use, and USDA's storage is research oriented. A combination of the problems faced at both facilities fueled the development of our system.

At Cargill, the products are stored in large silos, and the overall goal is to ensure that the stored products are kept at premium quality until distribution or processing. Guidelines in storage conditions are enforced by the Grain Inspection, Packer and Stockyards Administration (GIPSA) and the American Society of Agricultural Engineers (ASAE) (Lewis, 2006). These guidelines include equilibrium moisture content and upper bounds on temperature and relative humidity. However, to check these conditions, the workers at Cargill rely on random sampling with hand held sensors. Given that silos range in volume from 2350 ft^3 to 1,749,500 ft^3, it is impossible for such a sample to yield an accurate representation of conditions inside the entire silo. This technique also makes it difficult to respond to deteriorating conditions in their early stages. Yet another drawback of the technique is that it provides no data storage. This means that no historical data is available for querying or analysis. This explains why our correspondent from Cargill explained that they have no data management issues, no data. Therefore, after communication with Cargill two problems were unearthed, data acquisition and data storage.

USDA stores unshelled peanuts in mini-domes to simulate what occurs in silos and to develop proper preservation techniques. Unlike at Cargill, there is a monitoring system in use, and it controls the aeration within the mini-dome. Data storage is even implemented to preserve historical data for analysis. However, Microsoft Excel spreadsheets are used as the mechanism of storage; and they are not user-friendly for querying historical data. Much manual search time is exhausted to retrieve information from the spreadsheets. So, to add to the two previous problems of data acquisition and data storage, data management became the third area of inefficiency observed.

After communication with contacts from both facilities, we gained a better understanding of the problems faced with grain and seed storage. However, not only were we fully acquainted with the problem, but we were also educated to the background of the whole situation. An overall understanding of the importance of efficient grain storage was gained, and the significance of our system, once completed, was revealed.

1.2 Background and Problem Domain

Why is it so important to store cereal grain and oil seed efficiently? Anthropologists date the use of granaries (repositories for grain) and food stores back to ancient Egyptians as early as 4500 B.C. (Levinson, 1994). However, even the Egyptians were plagued with product degradation. The spoilage of their stored grains and foods sparked an overwhelming epidemic with insects. It is astonishing that now, some 6500 years later, there are still complications with preservation of oil seed and cereal grain products while in storage. Just to add clarity to the categorization of products, the group of cereal grain refers to barley, corn, oats, sorghum and wheat; and the group of oil seed refers to canola, peanuts, safflower, soybean and sunflower. Cargill and many other commercial storage facilities deal with these products daily.

Cargill is a family-owned food business that was started by William Wallace Cargill in 1865 at the end of the American Civil War (Broehl, 1992). It began with only one grain storage warehouse located in Conover, Iowa. Now, Cargill has grown into a global corporation with over 149,000 employees in 63 countries. Over the years it has evolved from just being a grain and seed repository into a commodity processing powerhouse (Broehl, 1998). Soybean is stored for processing into meal, oil and even vitamin E, and corn is stored for processing into ethanol, fructose, and even renewable products such as plastics and fabric. Most of Cargill's grain and seed products are stored in large silos, large cylindrical structures usually metallic or concrete (Beedle, 2001). While in storage, it is imperative that the products' quality be maintained before distribution to consumers or processing.

This is where USDA's research comes in. Facilities within the ARS, like the peanut lab in Dawson, are constantly conducting research to improve upon storage practices world-wide. While grain and seed products are in storage, they are susceptible to mold and deterioration. Conditions that are conducive for such events include high temperature, high humidity and high moisture of the product. This is why it is important to monitor the conditions of a stored product consistently. It is easy for the human eye to spot molding of a product; however, this is not feasible within a silo of such great volume. Therefore, research is done to discover the by-products of mold and deterioration so that it is identifiable just by observing conditions within the stored product. Figure 1.1 shows an example of deterioration within a corn sample.

Figure 1.1 Fungal Growth within a Corn Sample

Respiration is the process in which oxygen is utilized in the breakdown of carbohydrates, and then energy is released. When fungal growth occurs within the stored product, continuous respiration releases carbon dioxide (CO_2), water and heat. These three products can be used to detect deterioration in a stored product; however, of the three, temperature has been accepted as the standard method by the grain industry (Ileleji, 2006). The process of respiration within stored products is a topic that has been researched since the 1920s. Temperature sensors, mainly thermocouples, have been used in deterioration detection mainly due to their practicality and low cost. Areas of deterioration within the stored product are referred to as hot spots due to their increase in temperature. The presence of carbon dioxide has proven to be a good indicator of product deterioration in studies of stored cereal grain and oil seed (Rukunudin, 2004). However, due to its laborious methodology for data acquisition (until recently) and limitations in equipment, its use in the grain industry has never been fully implemented.

However, a recent study shows that CO_2 sensors should be reconsidered due to their ability to detect product spoilage in its early stages. As mentioned earlier, temperature testing is the grain industry standard for detecting product deterioration. One drawback of this method is that noticeable hot spots within a stored product are usually indicative of advanced stages of decay. Also, because of the low thermal diffusivity of stored grain or seed, a temperature reading would have to be taken within 0.5 m of a hot spot to detect it (Sinha, 1965). In the study (Ileleji, 2006), a hot spot due to fungal growth was simulated in a storage bin of corn. The fungal growth was initiated by automatically adding water to a controlled region within the storage bin. This region was equipped with five thermocouple (temperature) sensors, and a CO_2 sensor was installed in the overhead airspace of the storage bin. The experiment was repeated in three trials. In each trial there was a strong linear correlation between the rise in temperature recorded from the deteriorating region and the carbon dioxide concentration measured from the overhead airspace. Further testing revealed that CO_2 sensors were capable of detecting spoilage earlier than temperature sensors.

This occurs as a reversal of the initial use of CO_2 when applied to grain and seed storage. CO_2 has previously been used for fumigation to rid storage silos of insects. The normal concentration of carbon dioxide in the air is about 0.038% or 380 parts per million (ppm). During fumigation, the concentration of carbon dioxide in the air within the silo can be anywhere from 1% to 10% (10,000 to 100,000 ppm). These are levels at which all insects contacted would die, even humans. The use of this technique is not widespread because of its costliness, high maintenance and low rate of efficacy (CGC, 2004).

Research at the peanut lab in Dawson targets a common fungus in peanuts, *Aspergillus flavus (A. flavus)*. This fungus produces aflatoxin, a naturally occurring mycotoxin (Butts, 2006).

A. flavus is mainly found in crops after prolonged exposure to conditions of high humidity or even drought (Williams, 2004). Its native habitat is in soil, decaying vegetation and grain and seed undergoing microbiological deterioration. Control of this fungus is vital because aflatoxin is carcinogenic to humans, meaning it promotes cancer.

There are many incentives for grain and seed storage facilities to maintain the quality of their products before they go out to consumers or are processed. Besides the human health risk factor, fines and other corrective actions are in place to ensure that only high quality products enter the world's commerce.

1.3 Existing Techniques

Until recently, the main airflow through silos was ventilation. There were no elaborate aeration systems. For silos with only ventilation, when high moisture is detected within the storage product or deterioration is caught in its early stages, the product is shifted around. This is done by unloading and reloading the silo either partially or fully. Silos are unloaded from the bottom in the center and loaded at the top in the center. Sometimes this process is done randomly just to counteract the temperature gradient within the silo, given that the temperature of the product along the silo's walls fluctuates more than the temperature of the product in the center of the silo. However, it is often the case that this shifting of the stored product does not counteract the spoilage that may occur. With no aeration, spoilage and product deterioration is almost inevitable. Unfortunately, most of the time unloading just reveals deterioration that has already taken place (Ileleji, 2006). An example of such an occurrence can be seen in Figure 1.2. The owners of this silo unloaded the corn only to discover a molding area at the center. Standing mounds as such are also common in peanut warehouses where aflatoxin has attacked the peanuts.

Figure 1.2: Unloading Reveals Molded Mound

The most effective way of controlling conditions of products when stored in a silo is aeration. Aeration is the practice of forcing air through a stored agricultural product to control the temperature and moisture within (Butts, 2006). The majority of present day silos are equipped with aeration systems. The main setup involves the system being based at the bottom of the silo and having airspace at the top of the silo with an opening to let air in or out as the fan(s) is/are running (Wilcke, 1998). Newer silos come equipped with perforated floors to allow airflow from the aeration system. Usually air is pulled down through the silo during aeration. If the silo does not have a perforated floor, one can be installed. However, to save on labor and cost, perforated air ducts along the silo floor serve as an adequate substitution in small-scale and medium-scale storage projects.

1.4 Related Work

Monitoring systems for silos are few and far between; however, the following company has produced software related to the work involved in this thesis. In the late 1990's, Monitor Technologies LLC started production of SiloTrack™ PC-based inventory management software (Monitor, 2006). This company was founded by George Gruber in Port Sanilac, MI in 1958. In its present location of Elburn, IL, it thrives as a leading supplier of level, flow, particle emission and aeration instrumentation for the worldwide powder and bulk solids market.

SiloTrack is an application software that provides users with a flexible, graphical interface for the SiloPatrol® and Flexar™ sensor systems, both manufactured by Monitor. Both systems contain sensors that are deployed in silos to obtain inventory information such as weight, level, volume and aeration flow rate, if applicable. SiloTrack includes both a Server and Client version and thus has the capability of providing inventory monitoring and management to an unlimited number of users at an unlimited number of facilities. The server software has the direct interface with the sensors within the silo(s). It also establishes security and access limitations for each remote user. The client software provides remote users with access to the inventory monitoring system (SiloPatrol, 2004).

SiloTrack is a Windows based program and boasts simple and intuitive operation. It offers three languages: English, French and Spanish; and it affords a user-friendly monitoring environment for up to 128 silos. Figure 1.3 provides an illustration of the main display screen. On this screen, the user is able to visualize the latest status condition of each silo and its material contents. The user is also able to see sensors that are activated and their locations within the silo. The section on the left side of the main display screen is the silo details frame, Figure 1.4. When a silo within the main display screen is highlighted, its detailed information shows up here.

Within this frame, the user is able to see details such as silo type (shape), level of material within the silo, and the dimensions of the silo. SiloTrack allows up to five sensors and four alarms per silo. It can be setup to generate automatic reports concerning silo history and sensor diagnostics. It can also notify specific users via email or text-message when an alarm occurs within a silo.

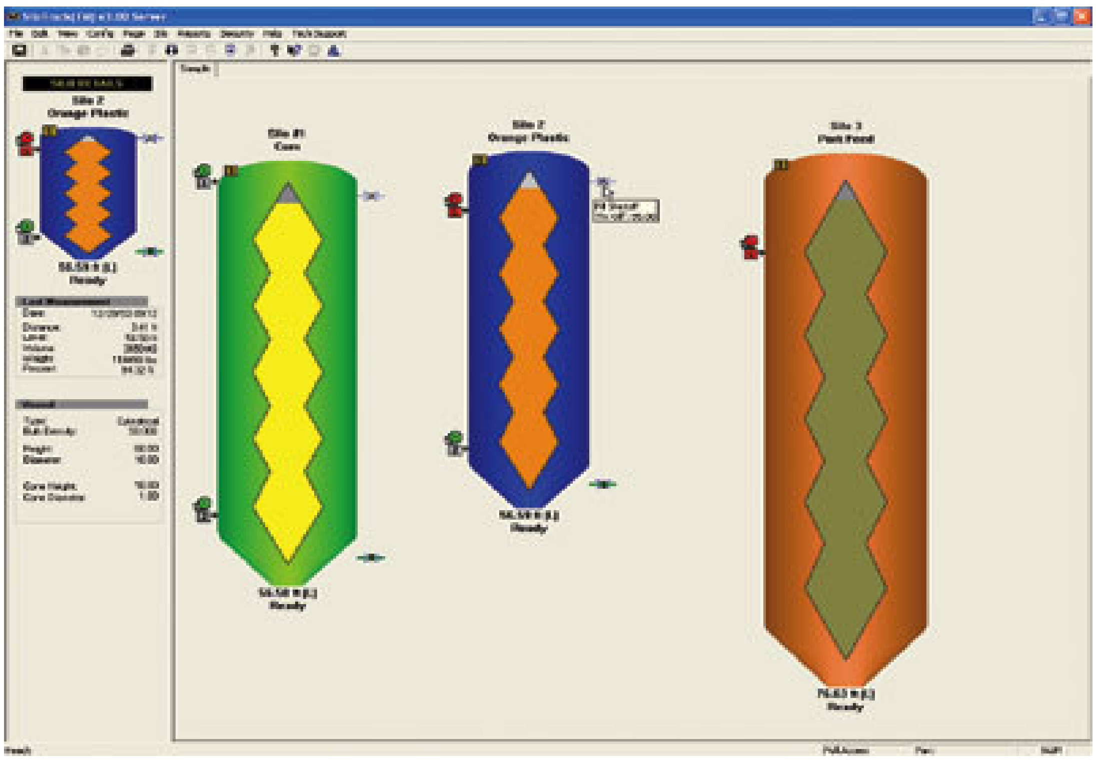


Figure 1.3: SiloTrack Main Display

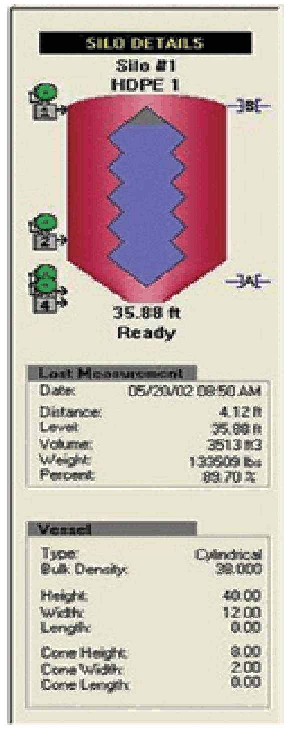


Figure 1.4: Silo Details Frame

The SiloTrack software provides a real-time monitoring system for silos, ideal for inventory control. The system in this thesis provides a real-time monitoring system for silos focused upon stored product preservation. The two projects correlate in approach and in problem domain. However, there are differences that serve as advantages for both systems.

The SiloTrack system excels in functionality. The user is able to configure the system basically to his/her liking and then monitor up to 128 silos. The networking capability of the system is also a plus. Users are able to gain remote access through LAN (local area network), WAN (wide area network)/Internet, or dial-up connections.

Our system excels in data management and storage. With the ontology in the background, the functionality of querying is provided within our system. SiloTrack does store historical data, and they can be viewed in the form of a chart. However, for further analysis and manipulation, the historical data have to be exported into either CSV, MS Word, MS Excel, or Lotus (SiloPatrol, 2004). Neither of these provides as easy a framework as the ontology for data analysis.

Parker et al (2006) discusses another software system, the GrainPlan™ Decision Support System, which is used by farmers and grain store managers in the UK. Grain stores are simply storage repositories such as barns, bins or bays used to store grain. This software is currently in its third version which was released in July 2006. The GrainPlan project was launched by the Home Grown Cereals Association (HGCA) in November 2002 under the Grain Sampling and Analysis Project (GSAP). This project was initiated to provide grain store managers with a tool to support the management of grain store quality (Parker, 2006).

GrainPlan offers a user-friendly environment for management of stored grain by monitoring temperature, moisture and even pest population. This software system utilizes modeling from years of historical data to predict events that will occur in grain storage based on current conditions. Upon initialization, the user is able to set up their site and place the grain stores (bins, bays, or heaps) as they appear in actuality. The user can simply click on an icon and drag it where he/she likes. This functionality is provided on the main user interface page, shown in Figure 1.5 on the next page. For each grain store, the user is prompted to enter pertinent details such as its size and contents. After this setup, the user may access screens for each individual grain store. An example of such can be seen in Figure 1.6. This page provides a three dimensional representation of the storage area showing points where data are recorded (these show up as small circles on the view page). These monitoring points are also positioned by the user. Just as the grain store setup, this procedure is only necessary at initialization. The user does not have to repeat setup of the site once configured the first time in GrainPlan. After every data collection, the user enters data recorded from each point: temperature, moisture, and/or pest population. After all data are entered, they are analyzed with the modeling prediction capabilities of the software, and each monitoring point is given a risk ranking: green being no current risk, yellow being some risk and red being high risk. The entire grain store's ranking is determined by the highest risk rank obtained from any of the monitoring points. For example, if a red risk is noted in even one monitoring point, the entire grain store is considered to be high risk.

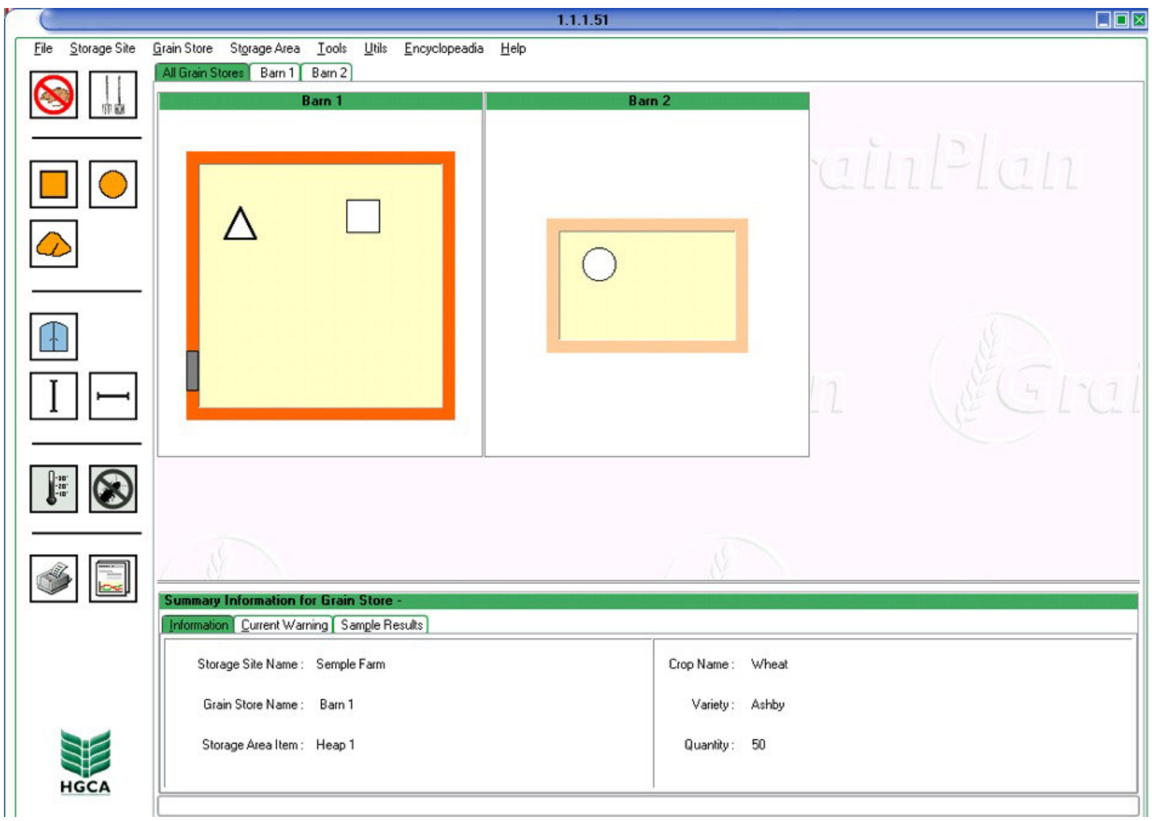


Figure 1.5: GrainPlan Main User Interface Page

Besides providing a risk color indication, GrainPlan also generates specific textual warnings explaining the source of the risk, the time limit before it becomes critical, and even the type of action to take. It even provides references to the warnings given so the user can fully understand the problem and how to resolve it (Parker, 2006).

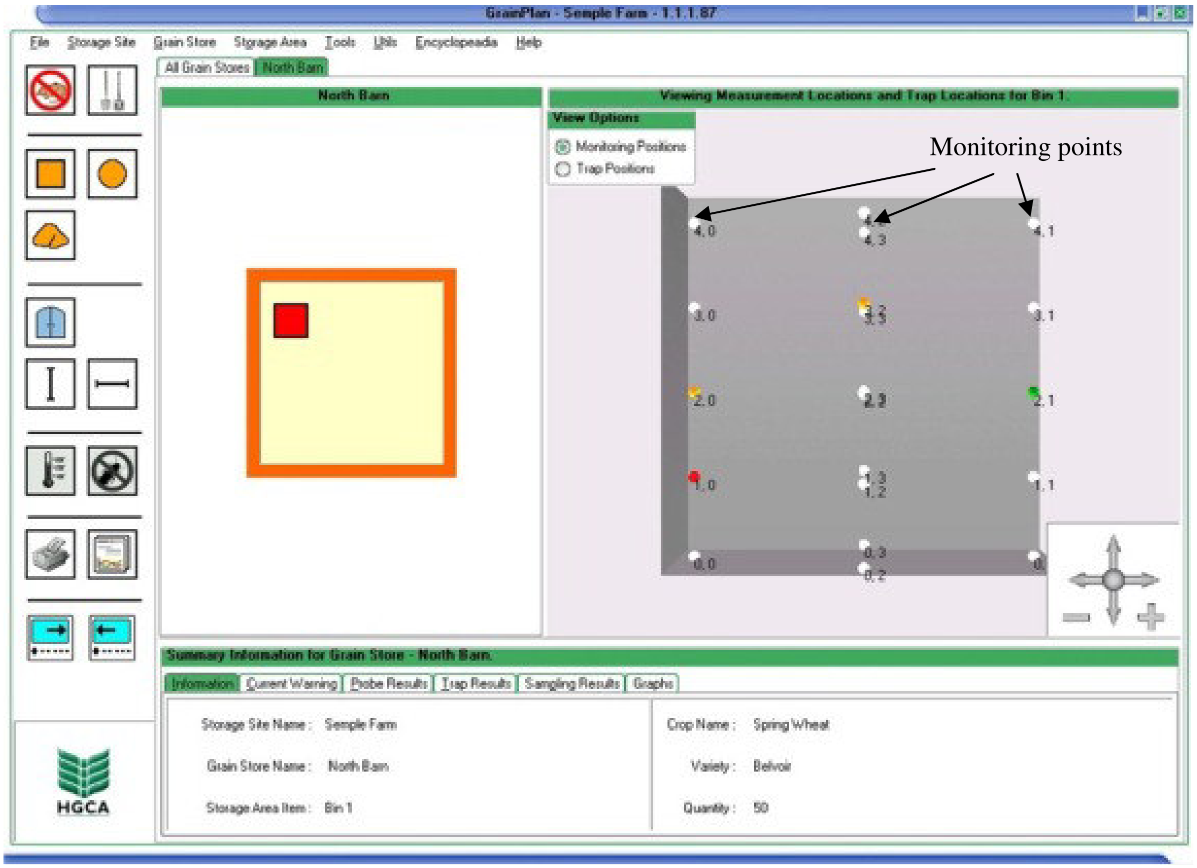


Figure 1.6: Individual Grain Store Page

After review of the GrainPlan Decision Support System, its functionality and practicality have to be recognized. It has been used widely throughout the UK ever since its first release in January 2004. It has been maintained and updated mainly through the feedback of its users. The main functionality it lacks now is the automation of data acquisition. As of now, the user has to manually enter or upload data to the GrainPlan system.

This next system also monitors grain and oil seed inside silos. However, instead of monitoring product deterioration, it monitors insect infestation. Surprisingly, losses due to damaged grain in the U.S. exceed $1 billion each year (Shuman, 2004), and the worldwide annual cost of protecting stored products from insect infestation is substantially greater. Present techniques for detecting insect infestations are expensive, labor intensive and require human entry to unsafe, confined spaces. Therefore, they are not often repeated, making insect control a scheduled event rather than one that is done based on the observed insect infestation. Scheduling presents the risk of insecticide overuse. When this is the case, insects develop a resistance to the insecticide, and the insect control becomes ineffective. The use of insect control is being “fine-tuned” globally by new governmental restrictions and mandates as a result of health and environmental concerns.

With these limitations on insect control, Shuman and Epsky (Shuman, 1999) developed an automated monitoring system to detect and count insects within silos. The Electronic Grain Probe Insect Counter (EGPIC) provides real-time monitoring utilizing infrared beam sensor technology from optoelectronic sensors. The data gathered is analyzed to determine population density within the silo, and the appropriate insect control methods are decided upon to counteract the insect infestation. The infrared sensors were strategically placed in traps throughout the silo. Each sensor unit consists of a transmitter and receiver. When an insect passes between them, the receiver detects that the infrared beam was broken in some matter. This is how EPGIC detects and counts insects. This system was rigorously tested to develop a threshold that would not count every small object (some could be grain especially if the kernel size is as small as wheat) but not ignore all small objects as well because some of them could be insects such as psocids and mites.

This system has since been improved with the addition of Sensor Output Analog Processing (SOAP) (Shuman, 2004). So instead of just counting the analog outputs from the infrared sensors, the system analyzes them. This enhancement eliminates erroneous counts and provides an indication of the species of the detected insects. This system targets four common stored-product pests: the flat grain beetle, saw-toothed grain beetle, red flour beetle and rice weevil. It is able to identify the insects by analyzing the amplitude of the analog output and the time duration for which it took the insect to fall through the infrared beam. The infrared beam has a cross-section of approximately 4.5mm. The largest of the four insects, the red flour beetle, has a length of about 4mm; and the smallest, the flat grain beetle, has a length of approximately 2mm. Each of the four insects was tested to observe analog output and time duration for them to pass through the infrared beam. Figure 1.7 shows a photograph of the infrared beam cross-section as compared to the silhouettes of the largest and smallest of the insects of interest.

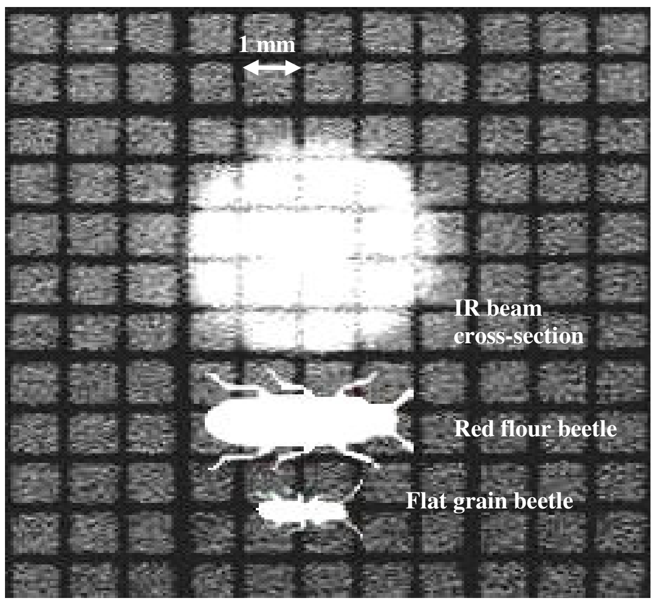


Figure 1.7: Photograph of Infrared Beam Cross-Section

So, here we have a system that utilizes the sensor network technology to create a real-time monitoring system within a silo or any grain storage facility. However, instead of it measuring environmental conditions such as temperature or relative humidity, its sensors are used to detect and identify living organisms, insects detrimental to the stored product.

The last related research we will discuss strays away from the concept of a sensor network enabling real-time monitoring within a concealed storage facility such as a silo or grain store. Instead, the sensor network is in an open field, reading atmospheric conditions and ground conditions. The Rural Broadband system provides real-time monitoring for grain and seed products before they are stored. In fact, it provides monitoring for crops in their earliest stages of production (Childs, 2006).

Camvera Networks and PreSoft Ag Solutions have joined to create a system to provide real-time monitoring for farmers to monitor their crops. This system was demonstrated for the first time at the Sunbelt Agricultural Exposition held in October 2006 in Moultrie, GA. The system is comprised of sensors that are deployed throughout the field. These sensors measure ground temperature, soil moisture, air temperature and relative humidity and report them back to a central station via a wireless cloud over the field, created by high speed broadband connectivity between the sensors and receivers. These sensors can be adjusted by the user to determine frequency of data readings.

The Rural Broadband system is forecasted to have a significant impact on crop productivity as well as reduce labor costs. The presence of streaming data would relieve farmers from physically having to go to the field frequently. The data readings would also aid farmers in preventing high stress conditions such as drought or over saturation for crops. The system can also be setup to autonomously control irrigation systems based on soil temperature and moisture readings gathered. It is currently in its early stages and is quite expensive, but farmers believe grants will help defray the cost of the system in the near future.

Each of the previously discussed systems exposes the use of sensor networks to implement real-time monitoring in agriculture applications. Whether it is for product preservation while in storage, inventory monitoring, or crop monitoring, this is a growing research area of interest. In the following chapters, we will discuss the development and functionality of our system and how we go about solving the stored product spoilage problem.

CHAPTER 2

SOLUTION APPROACH

With the three problematic areas in mind, data acquisition, data storage, and data management, we embarked upon the design of a system that would collaboratively address each issue. Though USDA only suffered from inefficiency in data management, we kept in mind that USDA and Cargill were merely two examples of many cases present in the world. It is highly likely that there are other corporations such as Cargill, small scale and large scale, which have inadequate provisions for product storage. Therefore, it was decided that the system would be given functionality that would suit most, if not all, candidates.

To solve the primitive data acquisition issue, a real-time monitoring system would be implemented to work in parallel with a sensor network. The sensor network would include sensors inside and outside the mini-dome to observe atmospheric conditions as well as conditions within. The data would be available for viewing by a user in real-time, and streaming data would be used to analyze conditions within the mini-dome to determine if any corrective action is needed. With the real-time system in place, no human interaction would be needed as far as sampling the stored product. This also initiates a preventative rather than reactive approach to product preservation.

The issues of data storage and data management would both be pacified with the use of a distributed dynamic ontology. As streaming data would enter the system, they would be stored persistently in the ontology. The ontology is described by the term “dynamic” because as the system runs, the ontology will grow indefinitely with the addition of more and more data.

Because of this storage technique, data would be much more manageable than it is in spreadsheets. Queries would be able to be facilitated, returning single results or multiple results. This would serve as a major advancement over the Excel spreadsheets.

2.1 System Design

At the conclusion of our brainstorming, we had a system consisting of four major components: the sensor network, ontology, data analysis and query processing unit, and the Graphical User Interface (GUI, user front end to facilitate querying and monitoring). Figure 2.1 shows how the four components interact within the system as data flow between them. The black arrows show the flow of data as they come from the sensor network, and the red arrows show the flow of data when a query is invoked. As data are retrieved from the sensor network, they first enter main memory as raw data. Then, they undergo analysis in the data analysis and query processing unit. Here, the data are analyzed to determine the conditions within the mini-dome. This reveals whether aeration is needed; and if so, what combination of fans. A full description of the sensor layout and a schematic of the mini-dome and aeration system are given in the third chapter in section four. When analysis is complete, the data go to the GUI for viewing by the user and into the ontology for persistent storage.

When a query is initialized by the user, the information goes to the data analysis and query processing unit. At this point, it is determined what files will be needed to fulfill the query. This request is passed to the ontology, and the needed files are returned to main memory. The data from these files are analyzed in the data analysis and query processing unit until the query is totally fulfilled. Finally, the query results are sorted and returned to the GUI for viewing by the user, and the recovered files are released from main memory back to the ontology.

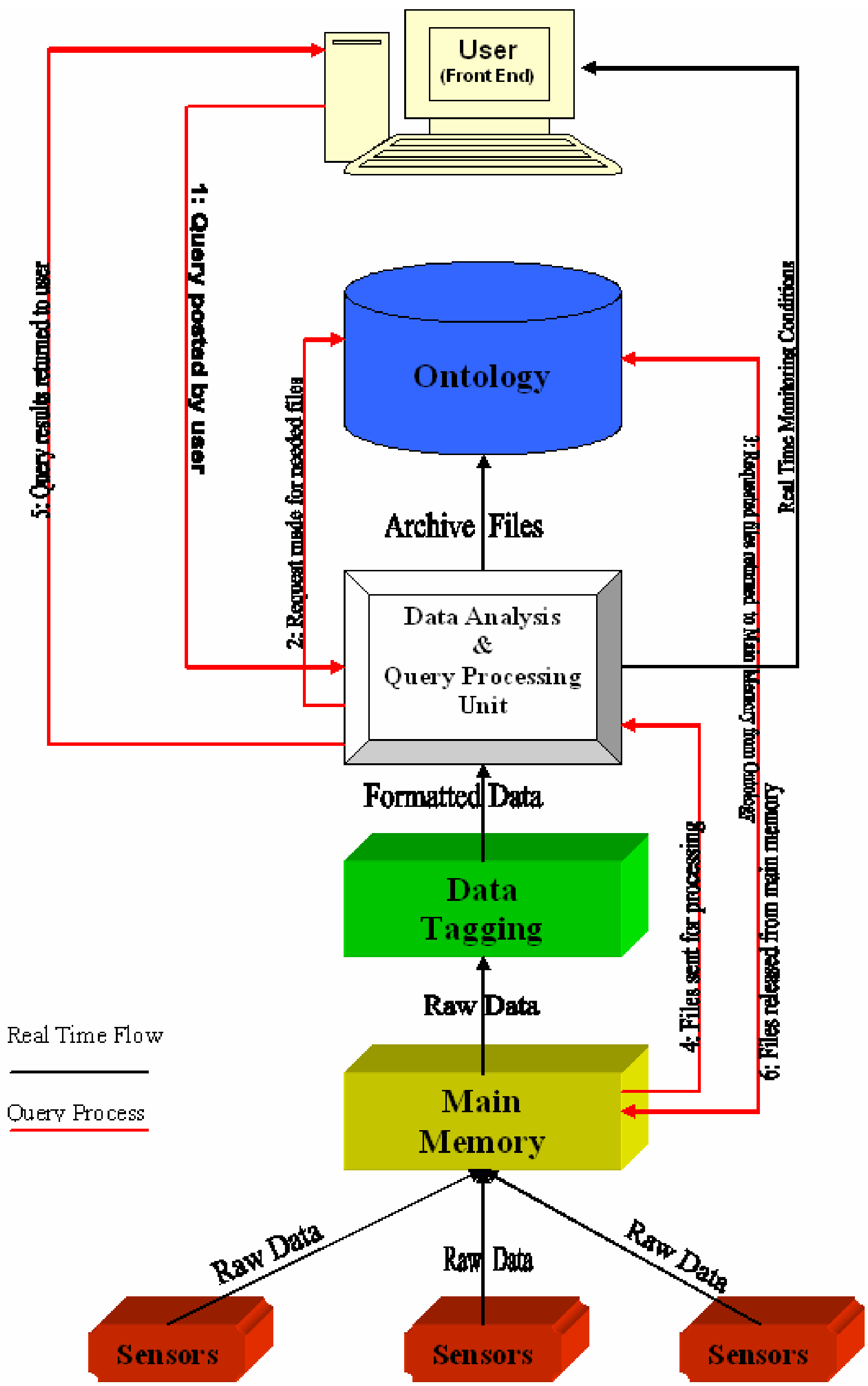


Figure 2.1: Complete System Diagram and Data Flow

The successful interaction between the four components discussed above and the seamless data flow between them results from the system's execution of six embedded tasks: data collection, memory caching, data tagging, ontology representation, query processing, and user interaction and data representation (Lewis, 2006). The successful completion of each of these tasks is imperative to the accuracy and efficiency of the system.

Data collection is a continuous process. Data are constantly streaming as long as the system is enabled. The initial collection of data occurs as they enter the system raw, directly from the sensors. The final residence for the data is in the ontology. Main memory caching is a performance enhancing task that was instated to preserve the efficiency of the system. In this task, records are cleared from main memory daily to prevent unnecessary clogging and ensure speed of data analysis and query response. The task of data tagging is extremely important and has to be executed flawlessly 100% of the time. It is here where data are labeled corresponding to the sensors they came from. The records are also time stamped with date and time to eliminate ambiguity.

The ontology representation is critical because a good setup allows for better data management and association discovery. The ontology serves two roles in our system; it contains grain-specific constraints (i.e. maximum humidity and maximum temperature), and it works as a persistent repository for all data. The ontology is discussed in detail within chapter four in sections three, four and five.

Query processing is a user-driven task. When a user invokes a query, it is in this process that the query is analyzed, and the determination is made how to fulfill the request. Then, the correct files needed are decided upon and returned to where they need to be. The appropriate results are then gathered and presented to the user in a time-sorted format. This interaction between the system and the user is a product of the sixth task, user interaction and data representation. A user-friendly GUI was created to facilitate queries from the user and display data as they are retrieved during monitoring. The monitoring page shows the conditions inside the mini-dome in real-time. It also shows fan operation and provides vivid color coded flags to alert the user when aeration is taking place. The query page allows the user to investigate certain aspects of the historical data.

CHAPTER 3

SENSOR NETWORKS

This system uses a sensor network to obtain data that represent the conditions inside the mini-dome. The sensor network establishes a real-time system for monitoring these conditions. The following sections of this chapter describe the features of sensor networks and provide a rationale as to why a sensor network was chosen as the mechanism for data acquisition.

3.1 Sensor Network Background

Here are two definitions that thoroughly explain what sensor networks are. The first definition is given by Professors Jim Kurose and Victor Lesser, who were both professors for the sensor network class (CSCI 791L) at the University of Massachusetts during fall 2003. Kurose and Lesser describe a sensor network as "… a sensing, computing and communication infrastructure that allows us to instrument, observe, and respond to phenomena in the natural environment, and in our physical and cyber infrastructure". The second definition, found online (Haenselmann, 2006), describes a sensor network as a computer network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. It is evident from both definitions that sensor networks are extremely useful in gathering information from the surrounding environment.

The early uses of sensor networks can be traced back to military applications. Since many research projects originate within the military, it is no surprise that this technology also has its origin there. During the Cold War, a network of acoustic sensors (hydrophones) was strategically located on the ocean bottom to detect and track quiet Soviet submarines (Chong, 2003). This network was known as the Sound Surveillance System (SOSUS). Since then, more sophisticated acoustic networks have been developed for submarine surveillance. Yet another sensor network emergence during the Cold War was a network of air defense radars that were deployed to defend the United States and Canada (Chong, 2003). Sensor networks such as SOSUS were vital in the research and advancement of technologies observed in today's modern sensor nets. These technological advances were sparked by research initiated by the Defense Advanced Research Projects Agency (DARPA). Two of the programs started by DARPA were the Distributed Sensor Networks (DSN) and the Sensor Information Technology (SensIT) programs (Chong, 2003).

The DSN program was initiated around 1980 to address problems such as data processing, signal processing, communication, sensor management, distributed computing support and tracking. The main focal point was to analyze data and signals to track mobile targets, ground-based or in the air. Researchers from institutions like Carnegie Mellon University (CMU) in Pittsburgh, PA and the Massachusetts Institute of Technology (MIT) in Cambridge, Mass. collaborated to focus on research areas under the DSN program.

The SensIT program was initiated to address two new key areas of interest. The first was to develop new networking techniques suitable for highly dynamic ad hoc environments (Kumar, 2001). The second was how to extract useful, reliable and timely information from the deployed sensor network. These areas were targeted after the realization that there would not always be time to deploy a strategically located sensor network. The ad hoc setup would be more practical; and therefore sensors would have to localize themselves and start retrieving data.

3.2 Uses of Sensor Networks

As the availability of low-cost sensors increased and advances were made in network communications, the use of sensor networks grew. As a result, sensor networks now have many other applications besides military. Here are a few examples.

With the latest threats in terrorism, sensor networks have been turned to for infrastructure security and counterterrorism applications (Hills, 2001). More and more critical buildings such as power plants and communication centers are being equipped with sensor networks. These networks consist of video, acoustic and other sensors to warn against incoming threats. Instead of the sensors being stand-alone, they collaborate to provide more accurate data and reduce false alarms. For example, let's say a nuclear plant has been warned of a possible attack by someone in a tractor trailer. All the sensors are spread out and have a threshold set for when to sound an alarm. Now, it is possible for the sensors to spike at certain times and give off false alarms. This was evident from working with the Lego Mindstorms Robotic Kit in the Introduction to Robotics course at the University of Georgia. So, to verify the presence of a tractor trailer (which would span several feet), several sensors would have to throw alarms. This collaboration of sensors serves a double duty. It discards false alarms and catches cases that are real alarms that might have not been caught by a stand-alone sensor (Hills, 2001).

Environment and habitat monitoring is also a recent application of sensor networks. They are considered practical in this realm because the variables to be monitored are usually distributed over large regions (Steere, 2000). The Center for Embedded Network Sensing (CENS), located in Los Angeles, CA, studies the response of vegetation to climatic trends and diseases with the use of environmental sensors (Charny, 2002). This center also identifies, tracks, and measures the population of birds and other species with the use of acoustic and imaging sensors. On an even larger scale, the System for the Vigilance of the Amazon (SIVAM) is a sensor network in the Amazon Basin that aids in three major areas: environmental monitoring, drug trafficking, and air traffic control (Jensen, 2002). It is sponsored by the government of Brazil and consists of radar, imagery and environmental sensors. The radar sensors are located on aircraft, the imagery sensors are located in space, and the environmental sensors are generally on the ground. Together, these sensors work to capture data in aid of the three previously listed tasks.

Sensor networks have also crept their way into the commercial industry as a means of lowering cost and improving machine performance and maintainability (Chong, 2003). Sensors are added to machines to monitor vibrations, wear and lubrication levels. These tactics help prevent machine failure or even catastrophic errors. Since sensors can be manufactured the size of buttons or even dust particles, they can be deployed in areas inaccessible by humans. Even in the case of machinery malfunction, the sensors can use data to generate diagnostic codes revealing the problematic area.

As technology advances, more and more sensors are being added to today’s cars (Gould, 2005). Radar sensors are used in the implementation of adaptive cruise control. When set, this system detects the distance to a vehicle ahead and its relative speed and keeps an appropriate distance from that vehicle. Radar sensor technology is also engaged when cars are in reverse by alerting the driver of unseen obstructions or revealing the proximity to seen obstructions. Video sensor technology is used to alert a driver that the car is departing its lane. This system has to differentiate between normal lane changes and street turns or a driver swaying due to drowsiness or some impairment. Sensors inside the car on the seats reveal the size and head location of passengers. This information is used to disable the air bag in the presence of a small child. Also, this information is used to create more personal restraint systems by inflating airbags to different sizes or pressure depending on the size of the passenger. Most popular throughout all models and makes, countless sensors are in place to aid with steering, braking, and traction. These systems aid drivers in extreme weather conditions.

An older application of sensor networks is vehicle traffic monitoring and control (Chong, 2003). Most traffic intersections are equipped with sensors either overhead or buried to detect vehicles and control traffic lights. Video sensors are used to monitor congested areas. However, due to the costliness of the implementation of these systems, they are usually only deployed at critical locations.

3.3 Sensor Nets in Real-time

The majority of implementations of sensor networks involve real-time applications. The addition of the real-time constraint adds operational deadlines from event to system response (Juvva, 1998). Sensor networks as such rely on fast data retrieval and analysis to ensure efficiency of the system. In section 3.2, various applications of sensor networks were discussed, having the real-time description. The real-time constraint can be divided into two categories: hard and soft.

In a sensor network with a hard real-time constraint, the required response time is minimal, and it is absolutely imperative that the data are retrieved and analyzed within the operational deadline. Not only is it important for the operation to complete within a given time frame, but it is vital that the operation completes at all. With a hard real-time constraint, failure for the operation to complete in a timely manner or complete at all yields catastrophic results (Liu, 2000). In many cases “catastrophic” is defined by the user or person evaluating the situation. A failure in response time or completion of a task within a sensor network used to aid in driving stability and traction control on a car could cause a serious accident leading to injury or even death. Systems as such rely on fast and accurate processing of data gathered concerning the road conditions. This is merely one example of a sensor network containing hard real-time constraints.

On the contrary, a sensor network with a soft real-time constraint can still function with a late response or none at all, and the end result is not catastrophic. An example of this case is a sensor network deployed to provide surveillance of a location. A delayed response from one of the video sensors would degrade the overall video quality or cause skipping of a few frames, but overall it would still run.

Our system can be described as having both traits. The hard real-time classification comes from the fact that the system is relying on fast analysis of the conditions within the mini-dome to determine if aeration is needed, thus preserving the stored product. However, since the product takes more than a few minutes to deteriorate, a few missed readings would not be catastrophic to the system or the stored product.

3.4 Sensor Network: Mechanism for Data Acquisition

Now that a preface has been given on sensor networks, it should be easier to comprehend why this approach was taken to acquire data within our system. The realm of applications for sensor networks is broad, and we felt that this would be another great instance to exploit the usefulness they provide in gathering data from a surrounding environment, which in this case happens to be the interior of the mini-dome. This section reveals the configuration of the sensor network and its role within the real-time monitoring system.

The sensor network consists of 16 sensors inside and outside the mini-dome. These sensors collaborate to provide data that are analyzed to determine if aeration is needed within the mini-dome. If aeration is required, it is also determined which fans to operate. Three fans are located on the floor of the mini-dome. These fans are located in perforated ducts and perform aeration by pulling air down through the stored product, which in this case are peanuts. Figure 3.1 illustrates the schematic of the mini-dome as well as the sensor layout (Lewis, 2006).

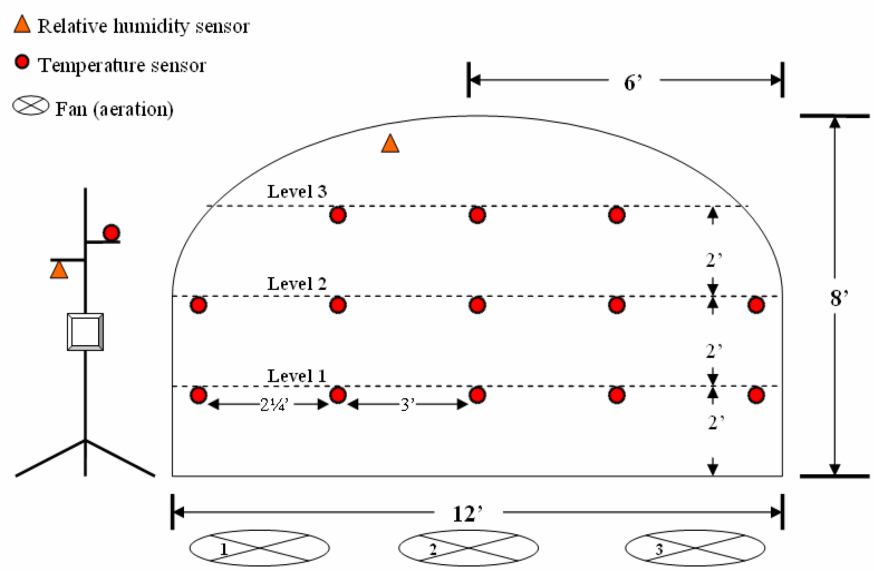


Figure 3.1: Mini-dome Schematic and Sensor Layout

Of the sixteen sensors, 14 are temperature sensors and the remaining two are relative humidity sensors. As shown in the diagram, the mini-dome has a height of 8 feet and a 12 foot diameter. Within the mini-dome, 13 temperature sensors are arranged in levels: Level 1, Level 2 and Level 3; they are spaced evenly two feet apart. Level 1 is located two feet from the floor and consists of five sensors. One sensor is placed at the center, and, in both directions, the next two sensors are placed 3 feet away from the center sensor. The outer two sensors are located 5¼ feet from the center sensor, making their distance from the wall nine inches. Level 2, also consisting of five sensors, is four feet from the floor, and the sensors duplicate the layout of Level 1. Level 3 is six feet from the floor; but it only consists of three sensors. The layout is the same as Levels 1 and 2 except for the absence of the outer two sensors. This is mainly due to the curvature of the roof of the mini-dome. There is also a relative humidity sensor suspended in the airspace at the top. Outside the mini-dome, on a mini weather station, there is a temperature sensor and a relative humidity sensor to measure atmospheric conditions. These sensors are tagged ambient, denoting outside conditions.

Although the levels of sensors are laid out horizontally, the sensors are grouped vertically for data analysis. Fan 2 is controlled by conditions evaluated from the center sensor in each level, thus representing the center of the mini-dome. Fan 1 is controlled by conditions evaluated from the left side, which (counting from the left) consists of sensors 1 and 2 from Levels 1 and 2 and sensor 1 from Level 3. Fan 3 is controlled by conditions evaluated from the right side, which consists of sensors 4 and 5 from Levels 1 and 2 and sensor 3 from Level 3. Within the data analysis and query processing unit, the temperatures of the left, right and center sections as well as the mini-dome as a whole are averaged using the four following equations:

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

$$centeravgTemp = \frac{Level_1 Sensor_3 + Level_2 Sensor_3 + Level_3 Sensor_2}{\#\text{ of sensors in center section}} \qquad \textbf{(2)}$$

$$leftavgTemp = \frac{\left(\sum_{i=1}^{2}\sum_{j=1}^{2} Level_i Sensor_j\right) + Level_3 Sensor_1}{\#\text{ of sensors in left section}} \qquad \textbf{(3)}$$

$$rightavgTemp = \frac{\left(\sum_{i=1}^{2}\sum_{j=4}^{5} Level_i Sensor_j\right) + Level_3 Sensor_3}{\#\text{ of sensors in right section}} \qquad \textbf{(4)}.$$

The fan operation criterion involves comparing the calculated average temperatures to the ambient temperature outside. If the average temperature of the mini-dome is greater than the ambient temperature, the fans are treated as a single entity, and all fans are turned on for aeration. If the average temperature is less than the ambient temperature, the mini-dome is evaluated in regards to the three sections, and the fans are controlled individually. The left side, center and right side are evaluated to determine the operation of fans 1, 2 and 3, respectively. There is only one scenario that can negate aeration, even if analysis determines it is needed. This is the case where the ambient relative humidity is 80% or higher. With a relative humidity this high, the air is very moist; therefore, (keeping in mind that the fans pull air down through the peanuts) aeration at this time would bring in moist air, causing more harm than help (Butts, 2006). With this being said, the fans are only operated if the ambient relative humidity is deemed satisfactory.

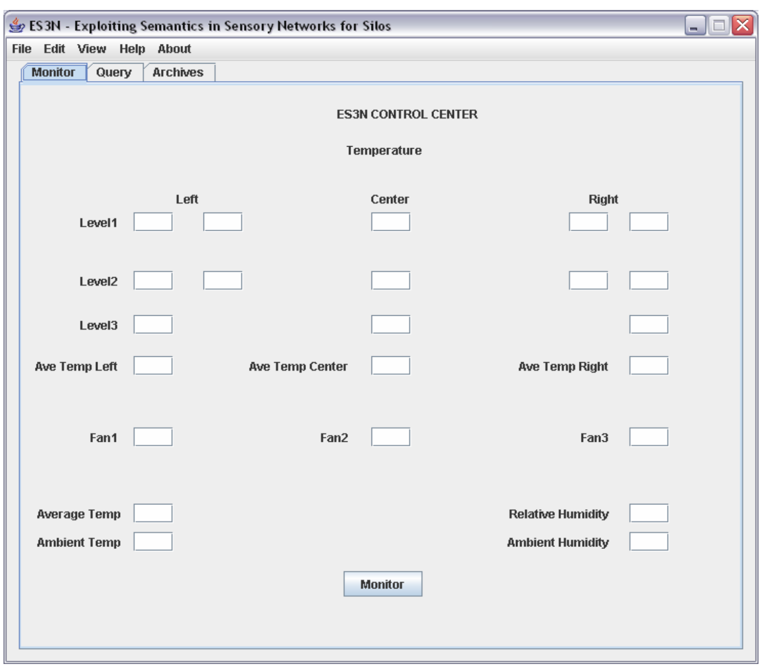


Figure 3.2: Monitor Page of GUI

In Figure 3.2, the monitor page of the GUI is illustrated. On this page, the data are intuitively displayed to the user providing a real-time, accurate view of the conditions inside the mini-dome. The display of the data is structured to resemble the layout of the sensors. The display updates hourly as data are retrieved from the sensor network and analyzed. Color-coded flags are thrown showing fan operation and trouble zones inside the silo. If one fan is on, as a result of one section having a higher average temp than the ambient temperature, the data for that section receive a yellow background, pinpointing the troubled region. If two fans are on, the data for those sections receive an orange background; this warrants more concern than just one fan running. If the decision is made to activate all three fans, a red flag is thrown notifying the user of this action. Rather than covering the whole page with red, it was decided upon to just change the background of the average temperature data display block. Figure 3.3 provides a view of the monitor page while active. This snapshot represents conditions inside the mini-dome on November 19, 2005 at 10:00 PM. In this case all three fans are in operation due to the average temperature of the mini-dome as a whole being greater than the ambient temperature and the ambient humidity being lower than 80%. However, one would also note that the center section is highlighted. This is because this section has an average temperature approximately 10°F higher than the ambient temperature, making it the major contributor to the mini-dome's average temperature being as high as it is. This warns the user to watch this region and take further action if conditions do not improve.

From background work as well as work presented in this thesis, it is evident that a sensor network serves as an efficient mechanism for data acquisition. It serves as an efficient way to lessen the human interaction needed to gather data; and it creates a robust monitoring system to guard against product spoilage.

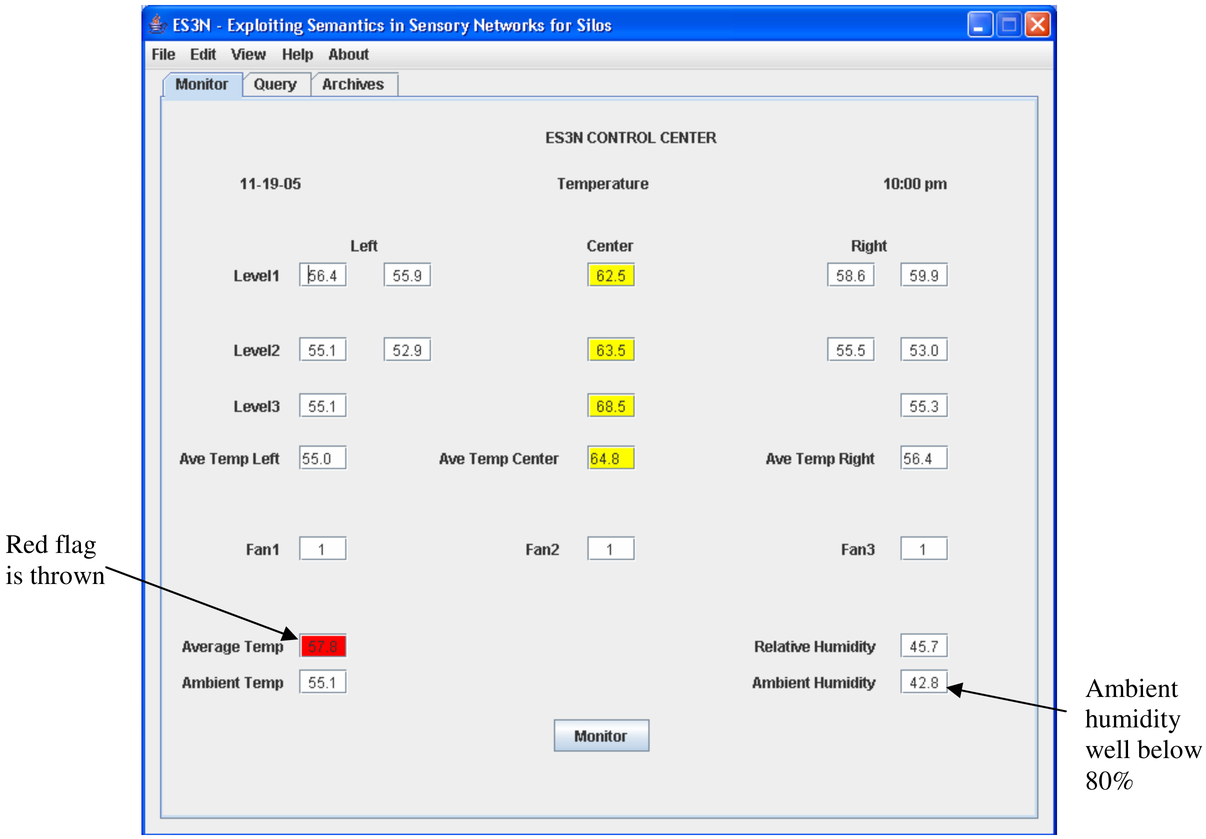


Figure 3.3: Active Monitor Page

CHAPTER 4

DATA STORAGE AND MANAGEMENT

The remaining two areas of interest in this thesis are data storage and data management. Chapter three illustrated how a sensor network was used to remedy the issue of data acquisition. In this chapter we will discuss the evolution of the Semantic Web and how it is proposed to emerge from the current World Wide Web. We will also discuss the benefits of Semantic Web techniques when applied to the storage and management of data. Hopefully the point will be conveyed as to why Semantic Web techniques were chosen as opposed to a conventional relational database.

4.1 Evolution of the Semantic Web

> "I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A 'Semantic Web', which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The 'intelligent agents' people have touted for ages will finally materialize."

This is a quote that was made by Tim Berners-Lee in 1999 as he expressed his vision of the Semantic Web. Sir Timothy John "Tim" Berners-Lee, born June 8, 1955 in London, England, is the inventor of the World Wide Web and the present day director of the World Wide Web Consortium (W3C) (Stewart, 2001). The World Wide Web is a global, read-write information space which provides access and cross references to text documents, images, multimedia and other items of information. W3C is an international consortium where hundreds of member organizations, a full-time staff, and the public work together in the development of standards for the World Wide Web. The mission of the W3C is "to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web" (Stewart, 2001). It was created to ensure compatibility and agreement among industry members in the adoption of new standards. It has received major support from DARPA, just as the sensor network research did, as discussed in chapter three.

The World Wide Web is considered to be the most significant computational phenomenon yet by many despite its chaotic development (Halpin, 2004). It has caused a major change in the way people communicate with each other and the way they conduct business. It is described to be at the core of a world-wide revolution that is presently transforming the world into a knowledge society (Antoniou, 2004). The World Wide Web's development has also gradually changed the overall perspective of computers. They are no longer seen as machines for merely computing numerical calculations; now they are predominantly used for information processing.

For the most part, the World Wide Web has been designed for direct human processing (Decker, 2000). If one would evaluate the Web, he/she would notice that it is typically used for seeking and making use of information, searching for and contacting other people, reviewing catalogs of online stores and ordering products by filling out forms, and transferring data and multimedia (Antoniou, 2004). The main tools used in these activities are search engines such as Google, Yahoo, and AltaVista. Search engines play a vital role in the use of the World Wide Web, but there are serious problems associated with their use (Bokor, 2000).

One problem faced by search engines is high recall and low precision. If a Google search is done for "the origin of the World Wide Web", the user is bombarded with approximately 31 million results. Although the relevant pages may be contained in these results, their presence is outweighed by the millions of irrelevant pages also returned. This is a case where too much is just as bad as too little. Many of the search engines rate pages and documents by hits instead of relevance. Therefore, search results will more readily return popular pages or documents in the subject matter than relevant pages or documents.

The opposite of the above problem is low or no recall. Although this occurrence is not frequent with current search engines, it still occurs. In this case, either a user submits a request that returns no results (very rare) or the results contain irrelevant pages.

Another problem is that results are highly sensitive to vocabulary. In my use of search engines, at times my search has been limited or taken a substantially long time just because of how the query was worded. Although meaning the same thing, queries worded differently can very easily return different results. This is insufficient because queries that are semantically similar should return similar results (Antoniou, 2004).

The last problem exposed within search engines is that results are single Web pages. This problem addresses the human interaction usually needed to traverse the results to uncover the needed information. Also, if the needed information is widespread over several pages or documents, it takes numerous queries to find them all.

Despite the advancements in search engine technology, the four above problems still persist. The assumption can be made that the technological progress is outpaced by the growing amount of Web content. The fact still remains that even in the case that the search is successful, it is the job of the user to browse the results and extract the information he/she needs. These problems and others are issues that will be remedied with the emergence of the Semantic Web.

The World Wide Web is currently based primarily on documents written in HyperText Markup Language (HTML). HTML is a markup convention that is used for coding a body of text interspersed with multimedia objects (Shadbolt, 2006). The shortcoming of this setup, as alluded to earlier, is that the Web content is not machine-accessible. It is mostly suited only for human consumption. The remedy to this shortcoming is the representation of content in a form that is more machine-processable containing metadata (data about data) and the implementation of intelligent techniques to take advantages of these representations. This proposal is known as the Semantic Web. It is imperative to understand that the Semantic Web will not be a new global information highway; it will gradually evolve from the World Wide Web (Antoniou, 2004).

The big change the Semantic Web provides is that instead of using HTML, it uses descriptive technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL). Together, these technologies provide descriptions that supplement or replace the usual content of Web documents (Fensel, 2002), causing the content to substantiate as descriptive data stored in databases accessible by Web or as descriptive markup within the pages or documents. These descriptions enable content managers to add meaning to the content of the web pages and/or documents, at the same time facilitating automated information gathering and research by computers.

The overall intent of the Semantic Web is to enhance the usability and usefulness of the World Wide Web and its interconnected resources through the four following tasks.

The first task is to provide documents that are “marked up” with semantic information (Fensel, 2002). This could be machine-readable information explaining the human-readable content of the document such as the creator, title and a description of the document. A person looking at an online catalog would be able to determine that the number below an item is its price; however, a computer would not be able to conclude such inferences without the presence of semantic information.

The second task is to create common metadata vocabularies (ontologies) and maps between vocabularies that would provide standards for document creators as they mark up their documents (Staab, 2004). This would make it feasible for agents to use the information in the supplied metadata. These vocabularies would also aid in diminishing ambiguity in semantic information. For example, "Author" in the sense of the author of the Web page would not be confused with "Author" in the sense of the author of a book that is the subject of a book review (Staab, 2004).

The third task is the provision of automated agents to perform tasks for users using the metadata provided within the Semantic Web (Antoniou, 2004). For example, humans are capable of using the World Wide Web to perform tasks such as reserve a library book, do comparison shopping, or find the Swedish word for "car". Before the Semantic Web, computers could not accomplish this task without human direction. However, with the appropriate metadata, automated agents, once deployed, could perform operations such as search for the cheapest DVD and buy it, find the nearest manicurist, or even book an appointment that fits a person's schedule (Walton, 2006).

The fourth and final task is the provision of web-based services to supply information specifically to agents. An example of such a web-based service is a Trust service that an agent could ask if a certain online store has a history of poor service or spamming.

The Semantic Web is propagated by the W3C (Antoniou, 2004). The catalyst behind the Semantic Web initiative is Tim Berners-Lee. Its development has major backing by industry and government. The U.S. government established the DARPA Agent Markup Language (DAML) Project in August 2000. The goal of this project was and is to develop a language and tools to facilitate the concept of the Semantic Web (DAML, 2006). The European Union's Sixth

Framework Programme (EUFP6) has made the Semantic Web one of its key action lines. EUFP6 serves as the framework for research, technological development and demonstration in the European Community (European Commission, 2002). It is a collection of the actions at the European Union level to fund and promote research. These and several other entities are working together to make the Semantic Web a reality instead of an abstract proposal.

4.2 Descriptive Languages: RDF, RDFS and OWL

RDF serves as the formal framework for metadata concerning the content in Web pages or documents. It is a simple language for creating assertions about propositions (Hayes, 2004). The basic idea of RDF is that resources can be identified and described in subject-predicate-object expression, referred to as a triple (Halpin, 2004). For example, in the sentence "The table has an oval shape", the resource, the table, is being described as having an oval shape. Therefore, the subject is "the table", the predicate is "has the shape", and the object is "oval". Figure 4.1 illustrates an example of an RDF graph that shows a representation for a person by the name of Eric Miller. The gray ovals in the graph are resources; and each has been assigned a specific Universal Resource Identifier (URI), a compact string of characters used to identify or name a resource (Shadbolt, 2006). The URIs that appear within the gray ovals (references) identify what the node represents. The URIs that are placed along the arcs (connecting lines) are used as predicates to identify relationships between the connected nodes. From this figure, we can see that the resource with the URI, http://www.w3.org/People/EM/contact#me, has the properties "fullName" and "personalTitle", which have the values Eric Miller and Dr., respectively. The other two properties, "type" and "mailbox" have values that happen to also be URIs, http://www.w3.org/2000/10/swap/pim/contact#Person and mailto:em@w3.org, respectively.

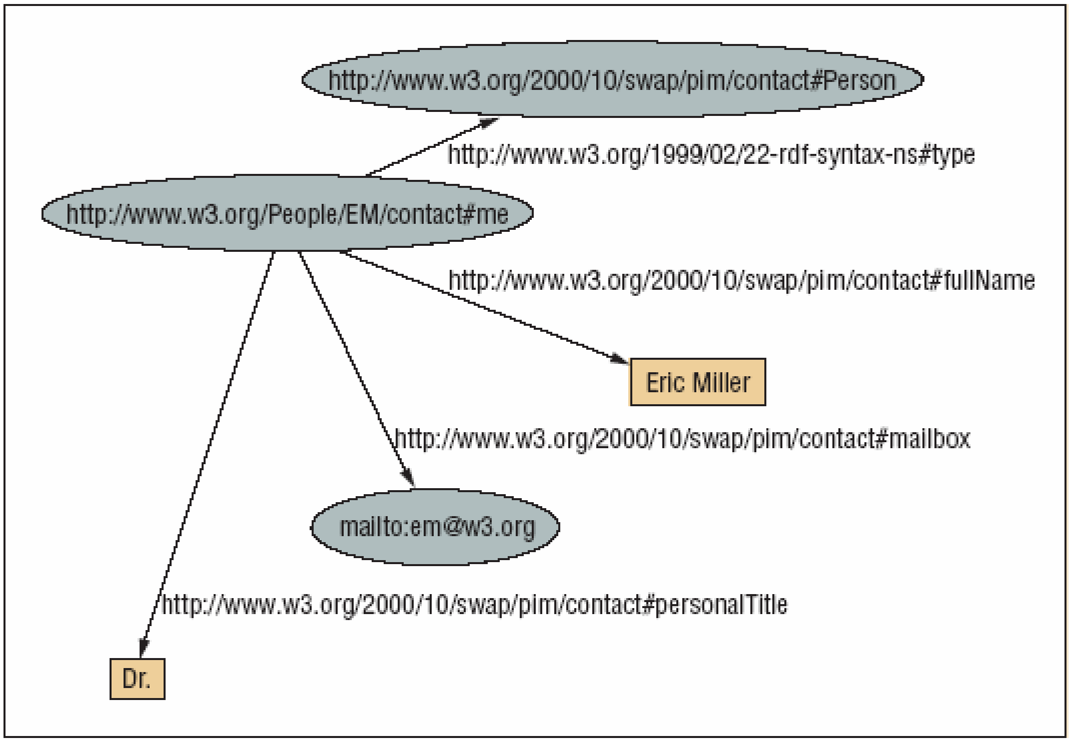


Figure 4.1: RDF Graph Representing Eric Miller

The following RDF snapshot shows how the information in this graph is written in a machine-readable format to be presented to a computer.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
         <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
            <contact:fullName>Eric Miller</contact:fullName>
            <contact:mailbox rdf:resource="mailto:em@w3.org"/>
            <contact:personalTitle>Dr.</contact:personalTitle>
         </contact:Person>
</rdf:RDF>
```

If this information was on a Web document, a human would be able to connect the properties “Dr.” and “em@w3.org” to Eric Miller. However, a computer would be unable to make such an inference. RDF provides a descriptive mark up for the document so that the computer is also able to process what a human sees when viewing the document.

RDF Schema (RDFS) is an extensible knowledge representation language that provides basic elements for the definition of onotologies. It is mainly used to define the schema within ontologies (Brickley, 2000). RDFS is a primitive ontology language that deals with the key concepts of class, subclass relations, property, subproperty relations, and domain and range restrictions (Antoniou, 2004). An example of a use of RDFS would be to define the data type for a value of a property. RDF and RDFS work together and compliment each other. RDF provides a definition for describing relationships among resources in terms of properties and values. However, it is the role of RDFS to declare these properties and define relationships between these properties and other resources (Brickley, 2000).

Despite the importance of the descriptive features provided by RDF and RDFS, it was discovered that the expressivity of the two was very limited (Antoniou, 2004). On November 1, 2001, the W3C created the Web Ontology Working Group chaired by James Hendler and Guus Shreiber, which was created to fulfill the need for a more powerful ontology modeling language (Heflin, 2004). Before the group was disbanded in May 2004, it was successful in defining OWL, a language that would go on to become standardized and broadly recognized as the accepted ontology language of the Semantic Web. The main purpose of OWL is to provide standards that provide a framework for asset management, enterprise integration and the sharing and reuse of data on the Web (Heflin, 2004).

OWL builds upon RDF and RDFS and has a similar syntax. However, it is incorrect to describe OWL as an extension of RDFS because of the capabilities it provides in description and expressivity. There are five major additions of OWL that standout: local scope of properties, disjointness of classes, Boolean combinations of classes, cardinality restrictions, and special characteristics of properties.

OWL offers the capability of defining the local scope of properties. In RDFS, setting the range of a property would mandate it for all classes. However, OWL allows range restrictions specific to certain classes. For example, we would be able to say cows eat only plants, while other animals may eat meat, as well (Antoniou, 2004).

Being able to make classes disjoint is also an addition of OWL. This allows the specification that classes cannot contain the same elements. An example of this would be a restriction within an ontology that no person can be a member of the "female" and "male" class at the same time.

OWL allows construction of new classes by using Boolean combinations such as union, intersection and complement of preexistent classes. An example of such is forming a new "person" class with a union of the classes "male" and "female".

Cardinality restrictions, yet another addition of OWL, place limitations on the number of distinct values a property can have. One example of this restriction would be to say that a person has exactly two parents, or an airplane has at least one pilot. Within our ontology, the cardinality constraint is used to show that a silo can contain at most one type of seed at a time.

Lastly, special characteristics of properties can be defined in OWL. Properties can be tagged as transitive, unique or even the inverse of another property. The "eats" and "is_eaten_by" properties would exist between cows and plants and be described as being the inverse of each other.

OWL is seen as a major technology for the future implementation of the Semantic Web. With the capabilities mentioned above, it is evident that OWL is a powerful descriptive language. It has been and will continue to be instrumental in the development of ontologies.

4.3 Ontologies

Noy describes an ontology as an explicit formal specification of the terms and concepts within a domain (Noy, 2001). It defines a common vocabulary for researchers that share information within a certain domain. Its structure includes machine-interpretable definitions of basic concepts within a domain as well as the relations among them. Ontologies are used in the Semantic Web to represent knowledge concerning the world or at least some part of it. They usually describe individuals, classes, attributes, and relations (Uschold, 1996), names given to objects within the ontology.

The individuals, also referred to as instances, are considered as the basic components of an ontology. They may include concrete objects (resources) such as people, places, and automobiles; or they may contain abstract objects such as numbers or words. Ontologies are not always initialized containing instances, but once populated they provide one of their general purposes which is providing a means of classifying instances.

Classes are abstract groups, sets, or collections of objects (Gruninger, 1995). They can contain instances, other classes, or a combination of the two. Examples of classes are vehicles, schools, fish and food. When classes are contained within classes, the ontology reveals a class hierarchy or subsumption relation. In this format, the more general classes are towards the top, and very specific classes are at the bottom. The following diagram shows a typical class hierarchy that would be found inside an ontology about brands of automobiles. The rectangles represent classes, and the ovals represent instances.

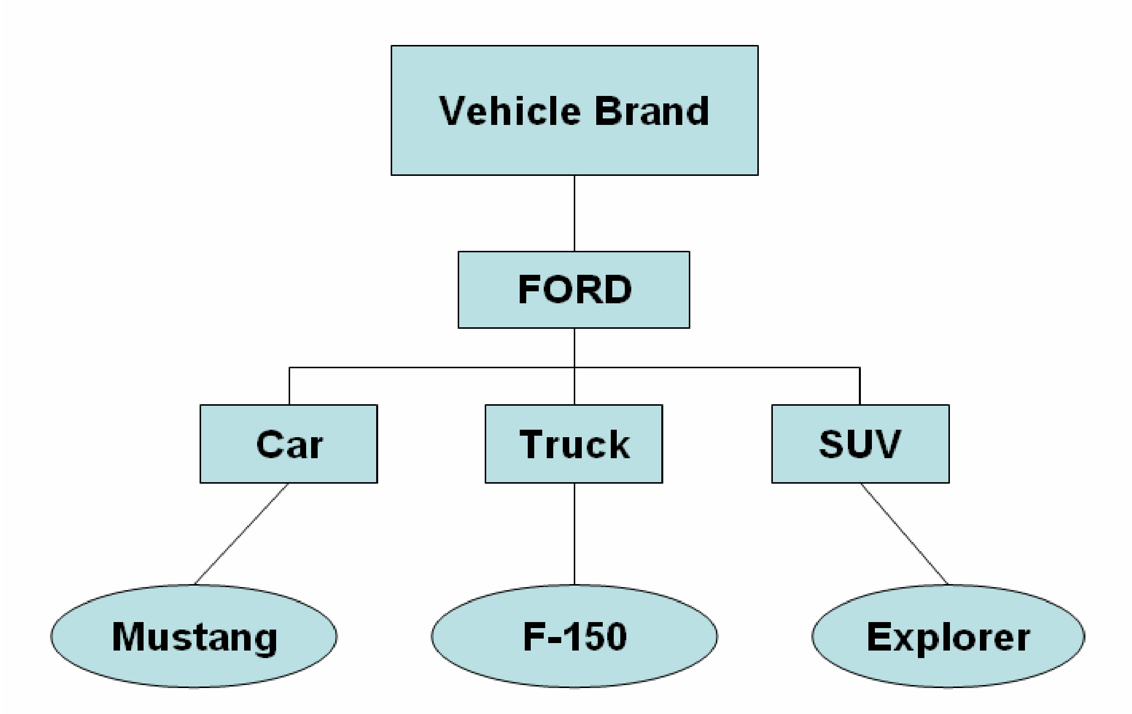


Figure 4.2: Class Hierarchy

Car, Truck and SUV are all subclasses of the Ford class. Ford is a subclass of the Vehicle Brand class. Mustang, F-150 and Explorer are instances of the Car, Truck and SUV classes, respectively. Ontologies provide these kinds of class hierarchies in many different domains.

Attributes are properties, features, characteristics, or parameters that objects can have and share (Noy, 2001). In this thesis, attributes are referred to as properties to maintain the context of the Semantic Web instead of that of a relational database. Within the ontology, objects can be described by assigning certain properties to them. The value of these properties can be data types such as strings, floats and integers. For example, the Ford Explorer instance in the above example could be given properties such as “number_of_doors” and “engine”.

Relationships are simply relations between objects in an ontology. The most important type of relation is subsumption (Uschold, 1996). This type of relation defines which objects are members of classes of objects. Adding “is-a” relationships to Figure 4.2 would create a taxonomy, a tree-like structure that clearly illustrates relations between objects. Within a taxonomy, each object is defined as the child of a parent class. Therefore, we would be able to deduce that Mustang is a car that is made by Ford; F-150 is a truck made by Ford; and likewise, Explorer is an SUV made by Ford.

It is believed that the growth and emergence of the Semantic Web is proportional to the provision of more and more ontologies (Shadbolt, 2006). For this process to accelerate, Shadbolt states that committed practice communities should develop, manage and endorse the ontologies that will furnish the semantics for the Semantic Web. There are ontologies already implemented in many facets of life (Decker, 2000). One such example is the Plant Ontology, developed by the Plant Ontology Consortium (POC), which represents objects in the domain of plant structures and growth/development stages. It serves as a complex hierarchical structure in which botanical concepts are described by their meaning and relations to each other. Another example is the Gene Ontology which was constructed in 1998. Since then, it has grown to contain over 19,000 terms that apply to a wide variety of biological organisms.

4.4 Data Storage, Semantically

The previous three sections of this chapter have presented a background of the Semantic Web, and they have shown the benefits of Semantic Web techniques. In this section, we will discuss how we used Semantic Web techniques to provide persistent storage within our system. Section one of chapter two lists and describes the four components of our system, one of which is the ontology, which is used as our mechanism for data storage. The ontology used in this system plays a double role. It contains constraints specific to various cereal grain and oil seed products discussed in section two of chapter one, and it stores the instances as streaming data enter the system (Lewis, 2006). To develop our ontology, we used Protégé 3.2, which is a free, open source ontology editor and knowledge-base framework developed and distributed primarily by Stanford University. We chose Protégé 3.2 because it allows the creation of an ontology schema that can be exported easily to OWL and RDF/RDFS formats.

Upon the initialization of the system, the ontology schema consists of a set of predefined classes, properties, and constraints. The schema is loaded into the system in the format of an owl file. The three classes in the ontology are Silo, Seed, and Data. The following segment of the ontology schema shows the initialization of the classes.

```
<owl:Class rdf:ID="Seed">
   <owl:disjointWith>
     <owl:Class rdf:ID="Silo"/>
   </owl:disjointWith>
   <owl:disjointWith>
     <owl:Class rdf:ID="Data"/>
   </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Data">
   <owl:disjointWith rdf:resource="#Seed"/>
   <owl:disjointWith>
       <owl:Class rdf:about="#Silo"/>
   </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Silo">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
       >1</owl:cardinality>
```

In this segment OWL and RDF are used to describe special features of the classes. The “disjointWith” description identifies each class as being disjoint with the other two, meaning they share no common elements. Towards the bottom, notice that the cardinality of the Silo class is set to “1”. This restriction mandates that each silo can contain at most only one seed type at a time.

The Silo class, reflecting the name of the storage unit this system was designed for, consists of three instances, silo_1, silo_2 and silo_3. These three instances were created in the earlier version of this project in which we were using multiple silos. However, limitations in real data caused us to downsize to one silo. The mini-dome referred to throughout this thesis is labeled as silo_1. The other silos remain to show the ease of expandability possessed by our system. Each instance in the Silo class has the property “has_Seed”. This provides a means of always knowing what product is in a specific silo. In the event that the stored product within a silo is changed, all the user has to do is change the value for the “has_Seed” property of that silo. The following schema segment shows the contents of silo_1 (mini-dome).

```
<Silo rdf:ID="silo_1">
   <has_Seed rdf:resource="#unshelled_peanuts"/>
</Silo>
```

RDF is used alone because the only entities here are the resources, property, and the value of the property. The resources here are “silo_1” and “unshelled_peanuts” because it has properties of its own. These three lines show that silo_1 has the property “has_Seed”, and the value of that property is “unshelled_peanuts”.

The Seed class consists of 11 instances, each of which has four properties to define the constraints specific to each grain or seed product. The constraints reflect standards established by the ASAE to govern storage of certain products (ASAE, 2002). These constraints deal with moisture, temperature, humidity and carbon dioxide. Although all constraints were not used in the final implementation, they are ready for whenever the system is equipped with the appropriate sensors. Here we see the constraints for unshelled peanuts, which is the product stored in the mini-dome.

```
<Seed rdf:ID="unshelled_peanuts">
   <has_max_temperature rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
```

```
    >72.0</has_max_temperature>
    <has_max_co2 rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.03</has_max_co2>
    <has_min_humidity rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.2</has_min_humidity>
    <has_max_moisture rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >8.5</has_max_moisture>
    <has_max_humidity rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >0.7</has_max_humidity>
  </Seed>
```

The properties for unshelled peanuts, just as every instance in the Seed class, are “has_max_temperature”, “has_max_co2”, “has_min_humidity”, “has_max_moisture”, and “has_max_humidity”. From this segment we can conclude that the ideal storage conditions for unshelled peanuts include a maximum temperature of 72°C, a relative humidity between 20% and 70%, and an equilibrium moisture content no greater than 8.5%. Also, carbon dioxide levels within the silo should not exceed 0.03% or 300 ppm. All values for these properties are identified as floats.

The Data class is the part of the ontology that grows dynamically. This class contains no instances initially, but it does have 21 properties for instances it will receive once the system is enabled. This section correlates to the attributes of a relation in a relational database model.

```
<owl:DatatypeProperty rdf:ID="has_fan2">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_time">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp15">
  <rdfs:domain rdf:resource="#Data"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_humidityout">
  <rdfs:domain rdf:resource="#Data"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_tempout">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp32">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp23">
  <rdfs:domain rdf:resource="#Data"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_date">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_humidityin">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Data"/>
</owl:DatatypeProperty>
```

The above segment is only a portion of the properties defined for Data resources, which will be instances of the Data class. The entire property listing for the Data class is available in the Appendix. The unique properties of each resource, similar to what would be a primary key in a relational database, are “has_date” and “has_time”. Only four of the temperature related properties are shown: one reflecting temperature outside (“has_tempout”), and a temperature from each level within the mini-dome. The format of the internal temperature properties is “has_temp[level][sensor#]” . Therefore, the property “has_temp15” reflects the reading from sensor five on the first level (refer back to chapter three, section four for a layout of the sensors within the mini-dome). Also, only one of the three fan properties is shown, “has_fan2”. Each of the three fan properties receives a value of either 0 or 1 to indicate if the fan is off or on, respectively, at a certain time. The remaining properties are “has_humidityin” and “has_humidityout”. These properties correspond to the relative humidity inside and outside the mini-dome. The values for the properties are defined as strings within the schema. They are cast to their appropriate data type accordingly within the system.

As data enter the system, they are tagged accordingly and reside in main memory temporarily. However, everyday at midnight main memory is cleared, and the data are written to a file with a name reflecting that day's date. This in turn creates our persistent storage. Figure 4.3 shows a view of the archive files within the ontology after a simulated 17-day run of the system.

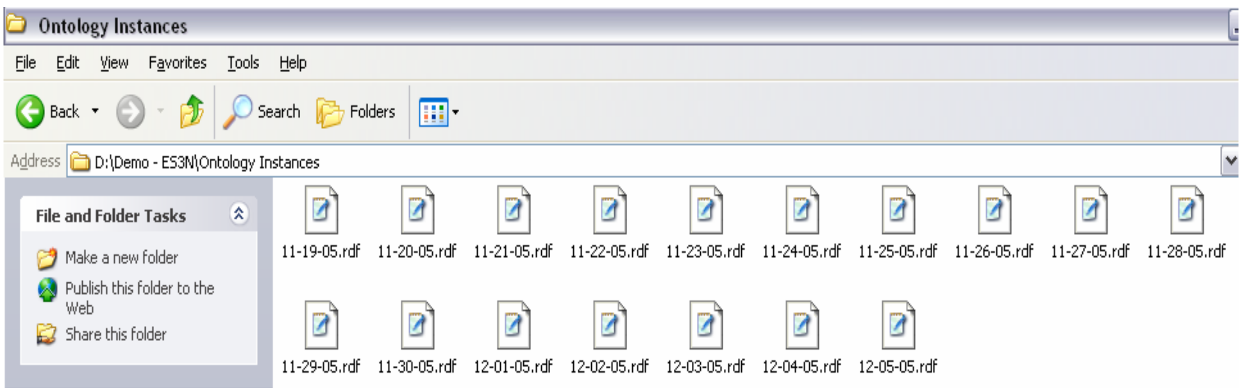


Figure 4.3: View of Archive Files within Ontology

The files are all written in only RDF because they only contain instances, the values for the properties identified above. Readings are performed every hour; therefore, every file contains at most 24 Data resources, each having 21 properties. These data resources and their properties can also be referred to as records in the ontology. Figure 4.4 illustrates one data resource and its properties from the "11-19-05.rdf" file. The data resource reflects 1:00 PM on November, 19, 2005.

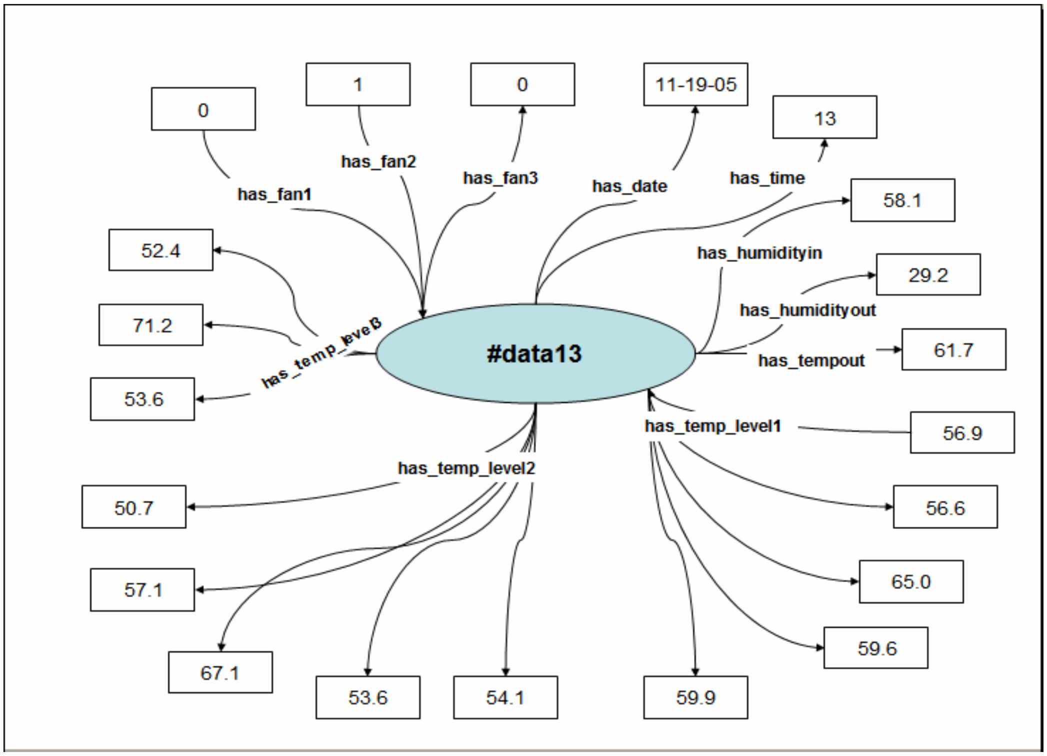


Figure 4.4: Illustration of Data Resource and Properties

The following RDF segment shows how the information in the above graph appears in the RDF file.

```
<rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1143053835.owl#data13">
   <has_temp23>67.1</has_temp23>
   <has_temp13>65.0</has_temp13>
   <has_temp33>52.4</has_temp33>
   <has_temp25>50.7</has_temp25>
   <has_temp31>53.6</has_temp31>
   <has_fan1>0</has_fan1>
   <has_temp11>56.9</has_temp11>
   <has_humidityin>58.1</has_humidityin>
   <has_fan2>1</has_fan2>
   <has_temp12>56.6</has_temp12>
   <has_tempout>61.7</has_tempout>
   <has_temp24>57.1</has_temp24>
   <has_humidityout>29.2</has_humidityout>
   <has_time>13</has_time>
   <has_date>11-19-05</has_date>
   <has_temp14>59.6</has_temp14>
   <has_fan3>0</has_fan3>
   <has_temp22>53.6</has_temp22>
   <has_temp32>71.2</has_temp32>
```

```
  <has_temp15>59.9</has_temp15>
  <has_temp21>54.1</has_temp21>
</rdf:Description>
```


Storing data semantically provides benefits that we uncovered during this project and could have even more benefits that we didn't come across. The overall benefit is the adaptability of the system provided by the semantic storage techniques. The class hierarchy and property definitions within the ontology allow for changes in the external entities without major internal software changes. For example, if the product stored inside silo_1 was changed from unshelled peanuts to wheat, the labor involved would be the unloading and reloading of the silo. The only software alteration would be to change the value of the "has_Seed" property of silo_1 from unshelled peanuts to wheat. All the constraints would already be present, and the flow and interaction would remain the same.

4.5 Meaningful Data Management

Recall that Cargill has no data management issues because they do not store any of the data; and USDA stores data in Microsoft Excel spreadsheets making data hard to manage. Our semantic data storage provides meaning to the data by exposing relationships and associations. We exploit this enhancement with the facilitation of querying of historical data. It will also be shown how semantic associations can aid in data analysis and mining.

First of all, after midnight of the first day that the system is initialized, data are stored persistently within the ontology. When a query is posted by the user, it is first analyzed to see which files will need to be returned to main memory for querying. These files are then retrieved from the ontology and brought back into main memory. Once the query has been satisfied, the files are released from main memory and return to persistent storage in the ontology. This process is shown in Figure 4.5. It is also illustrated in Figure 2.1 and discussed in detail in chapter two, section one.

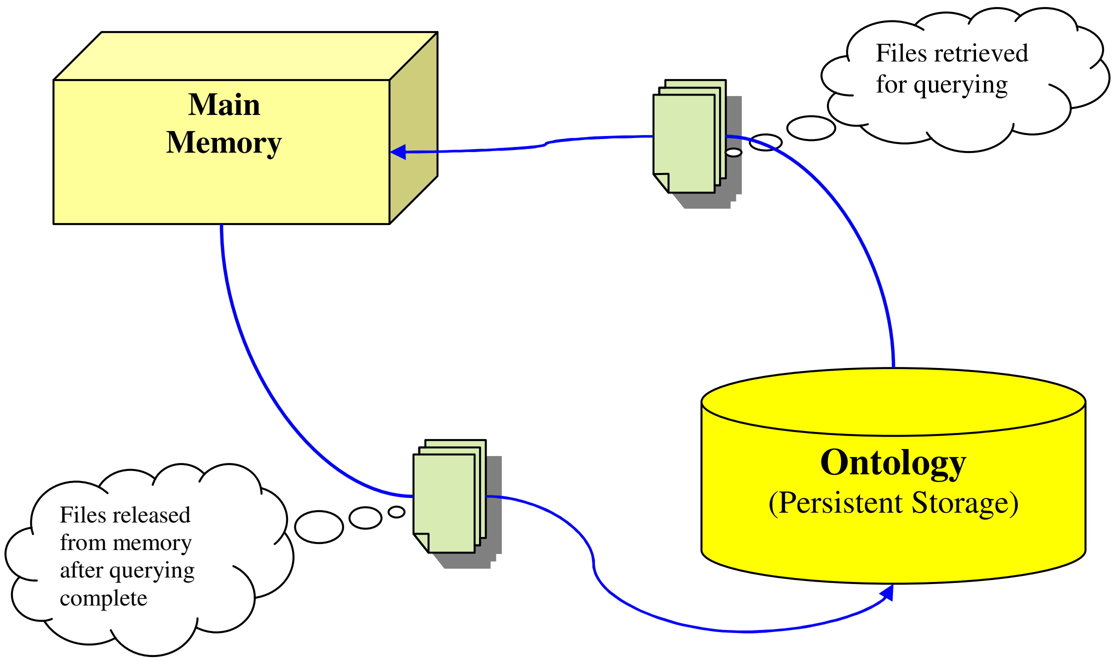


Figure 4.5: File Routing During Querying

Presently, our system provides facilitation for three types of queries: exploratory, monitoring and range. The exploratory query, the simplest type, retrieves a single record from the ontology. This would be the equivalent of the user selecting a specific date and time on the query page and submitting that request. The monitoring query searches and returns multiple records within one day; data from the same day would all be in one RDF file. This is equivalent to selecting a date on the query page and submitting that request. The range query is a query that spans data over a time period. For example, a query concerning when a specific combination of the fans were on would require the evaluation of data since initialization of the system. These three types of queries are reflected on the illustration of the GUI Query page shown in Figure 4.6.

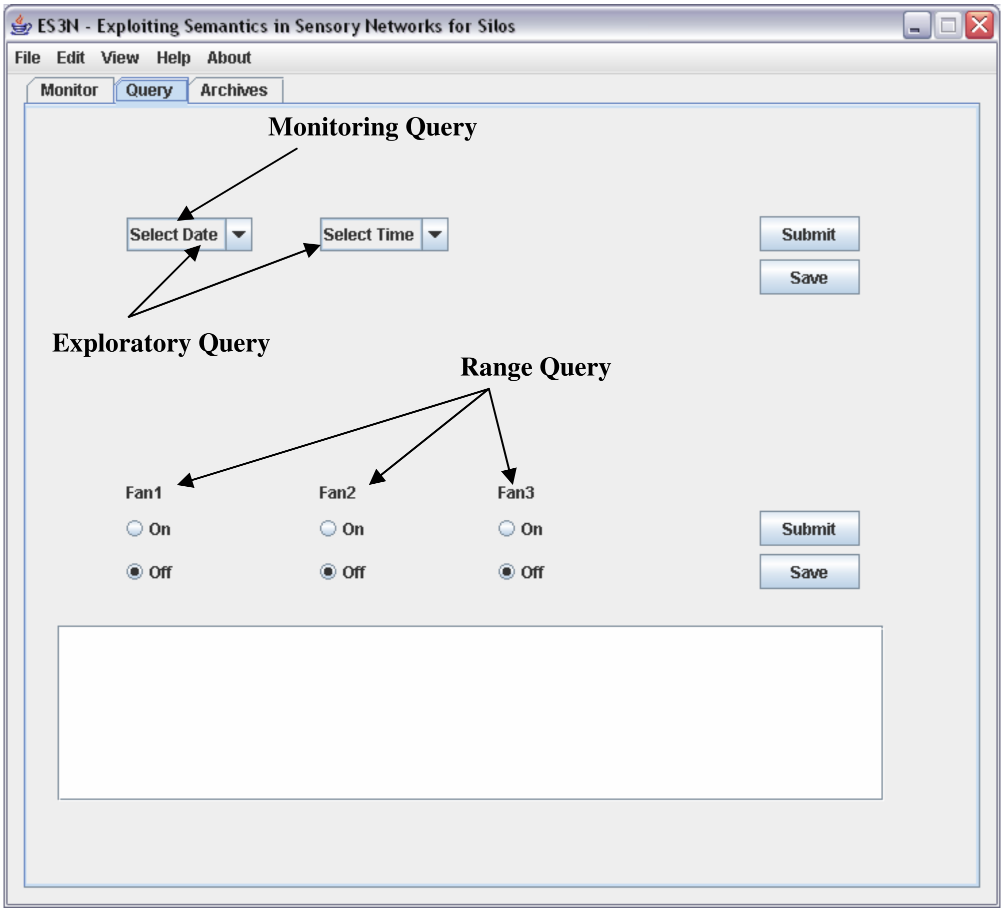


Figure 4.6: GUI Query Page

For our query processing, we use Simple Protocol and RDF Query Language (SPARQL), which was designed for accessing RDF data (Lewis, 2006). SPARQL is embedded in Jena, a Java framework for building Semantic Web applications and providing a programmatic environment for RDF, RDFS and OWL. Queries performed in SPARQL capture relationships among data triples and uncover semantic associations, as demonstrated in Figure 4.7.

In the following example we will discuss how semantic associations aid our system in query processing. Figure 4.7, provides an illustration of the example. There are four resources taken from the Data class of the ontology, each having 21 properties. For the sake of this example, only two properties are shown, "has_date" and "has_fan1". All four resources share the same literal value for the "has_date" property, "11-20-05". Three of the four resources share the same literal value for the "has_fan1" property, "1". These two cases show associations between the resources. Therefore, a query posted concerning the date of "11-20-05" would return all four resources because their "has_date" property points to the same value. Likewise, for a query concerning the "has_fan1" property having a value of "1", resources data1, data2, and data4 would be returned because of their association.

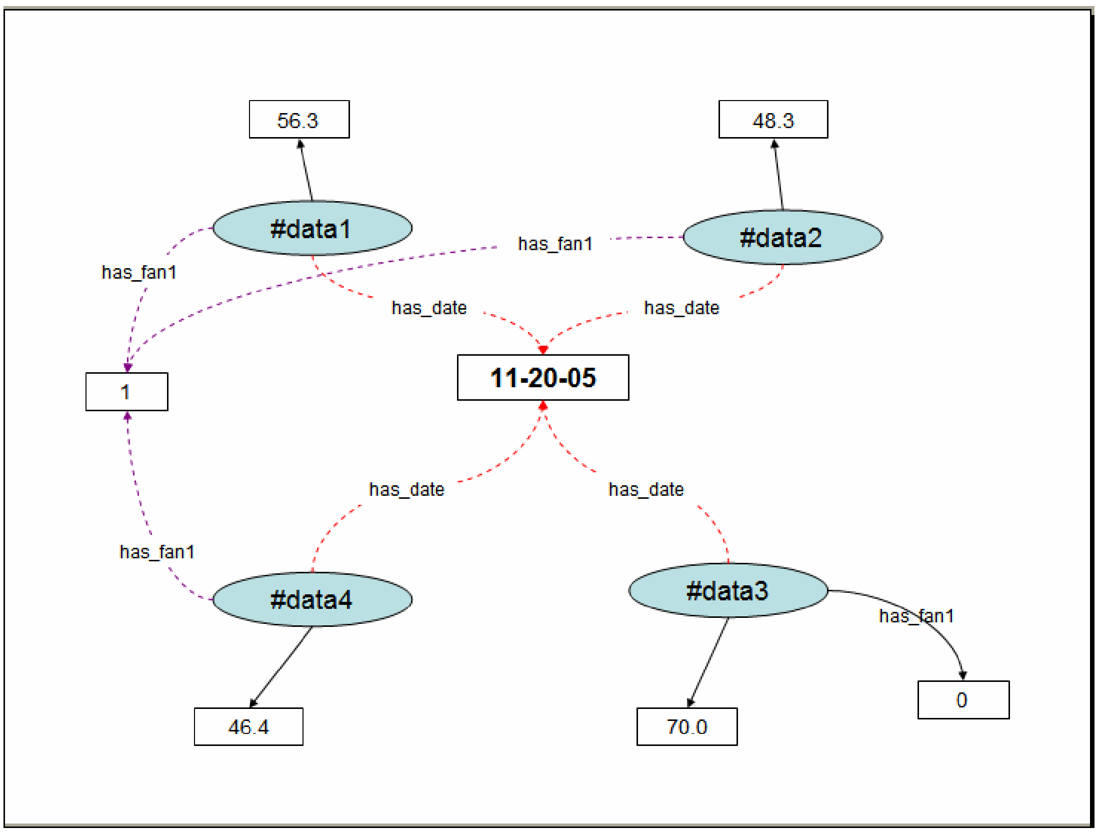


Figure 4.7: Semantic Association Aids Querying

The essence here is that literal values are being used to derive semantic associations between resources. This methodology is not available in relational databases simply because there, literal values are just that, literal values. Though it seems minuscule, this aspect is very important. These kinds of semantic associations point to the possibility for supporting much more complex queries that can traverse several hops, following many nodes and edges to a variety of endpoints. We are unable to exploit this finding within this thesis because the provided data lacks the depth to demonstrate such a complex association.

CHAPTER FIVE

RESULTS AND EVALUATION

We were successful in the completion of all three tasks proposed at the beginning of this thesis. The resulting system addresses the issues of data acquisition, data storage, and data management that plague the agriculture industry in the efficient storage of cereal grain and oil seed products. The sensor network and the use of Semantic Web techniques proved to be extremely beneficial in the development of a robust, automated monitoring system that provides data storage and facilitates queries within it.

The real data received from the USDA was used to test our system in simulation. It was not actually tested in the field, but it has been rigorously evaluated with actual raw data retrieved over a five month period. The first test of the system during development was an activation study, making sure that the fans' operation reflected the results and calculations made by the data analysis and query processing unit. This test turned out positive and confirmed that the data were being properly analyzed.

The second test performed compared our fan operation to that of the system present at the USDA lab in Dawson. This test was done to evaluate the correlation between the operation of our fans simultaneously and the operation of the single fan at the USDA lab. In the research at the USDA lab, there is only one fan that aerates the mini-dome. It is turned on if the average temperature inside the mini-dome is greater than the outside (ambient) temperature. This is the same rule that applies to the operation of the fans in our system all at once. Therefore, it would be expected to have a good correlation between the two systems in such a comparison. Several dates were chosen at random, and the fan operations from both systems were compared side by side. The correlation observed between the two systems was excellent. Figure 5.1 illustrates one day, November 19, 2005, when the correlation between the two systems was 100%. The overall correlation for the five month period was 91%.

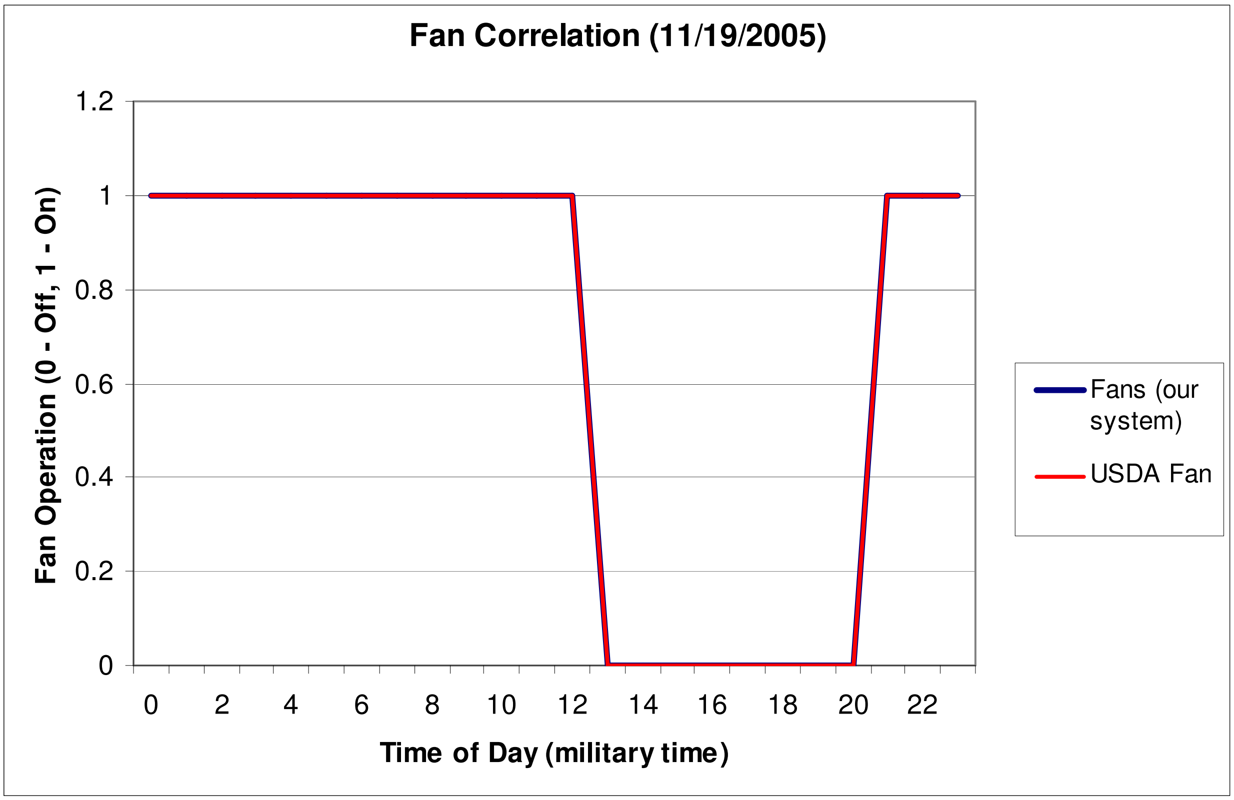


Figure 5.1: November 19, 2005 Fan Correlation

In the above figure, the fans from our system were operated simultaneously the same times as the fan at the USDA was operated. However, when data analysis shows that the average temperature of the stored product within the entire mini-dome is less than the outside temperature, the mini-dome is split into three sections (left, right, and center), and the average temperature of each section is evaluated (this entire process is discussed in chapter three, section four). This explains why in our system it is possible to have only one fan in operation or a combination of any two fans in operation. In Figure 5.2, each fan's operation from our system is shown individually versus the operation of the fan at the USDA. This graph shows variance in the operation of the fans from both systems.

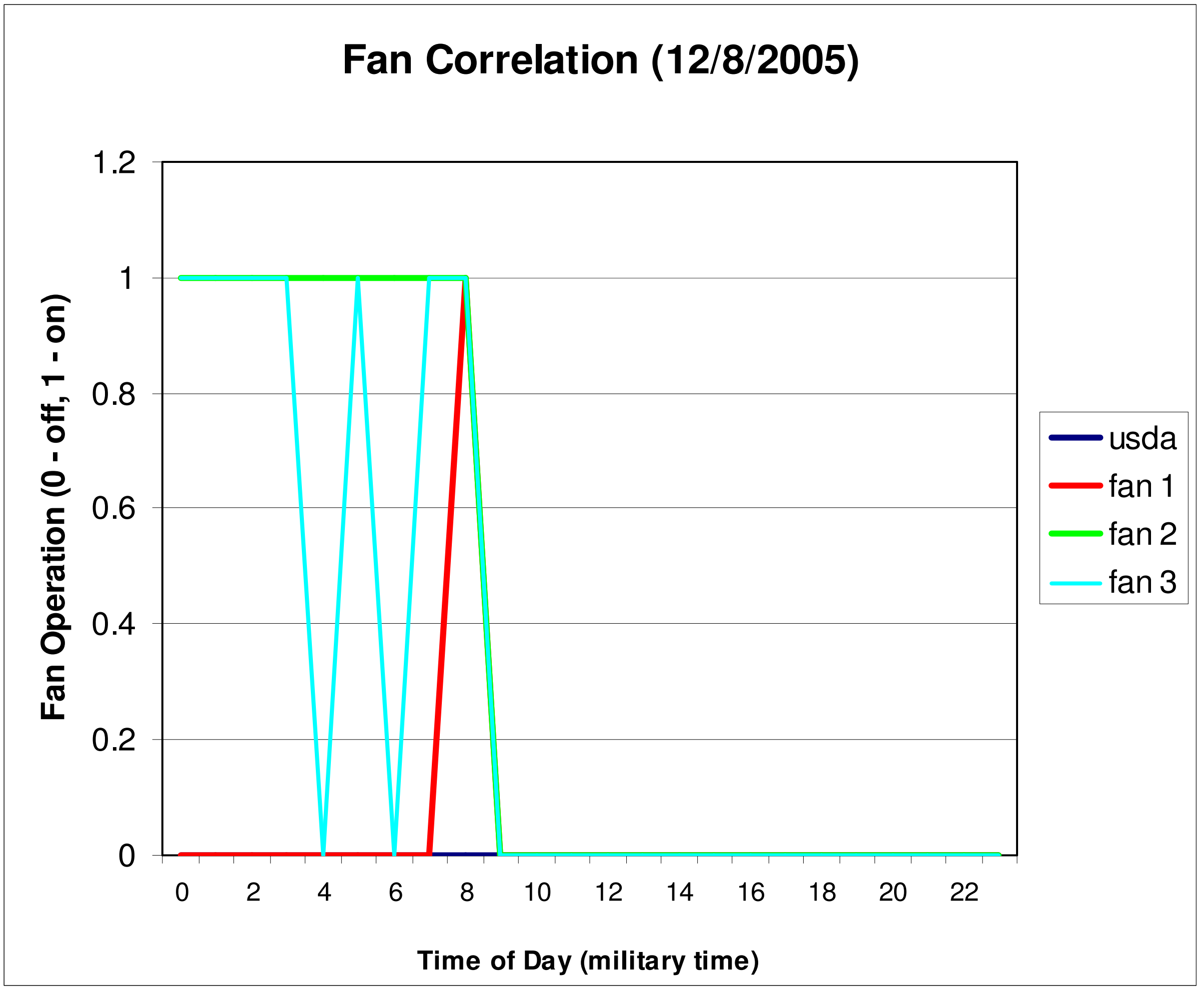


Figure 5.2: December 8, 2005 Fan Correlation

The most obvious detail shown in the above graph is that all fans shut off at 9:00 AM. This could be for two reasons: the relative humidity outside is over 80% (threshold to guard against bringing in moist air) or the average temperatures of the stored product within the entire mini-dome and the three sections are less than the temperature outside the mini-dome. The fan of the USDA system is turned off for the entire day. In our system, fan 2 (fan for the center section) is on at the start of the day and does not turn off until 9:00 AM. At times this fan is on by itself, signifying that the center section of the mini-dome had an average temperature greater than the outside temperature, but the average temperature of the entire mini-dome was still less than the outside temperature. Fan 1 (fan for the left section) is turned on at 8:00 AM and operates only for one hour. Fan 3 (fan for the right section) is on at the start of the day and is turned off at 4:00 AM. It is turned back on at 5:00 AM for one hour, and then it is turned back on at 7:00 AM for two hours before being turned off at 9:00 AM.

Times when one fan was on or two fans were on indicate that even though the average temperature of the entire mini-dome is below the outside temperature, there are still warm regions within the mini-dome. Sectioning the mini-dome provides a more detailed temperature observation and enables more efficient aeration. However, there was no time or the capability to test to see what impact the single operation of fans would have on the stored product within the mini-dome.

The third test was for query efficiency. In this test, the system was run in simulation over the full five month span of the real data. Each of the three queries were performed once after the first day and then on a monthly bases. The times were evaluated in milliseconds. Figure 5.3 shows a plot of the results from this efficiency test.

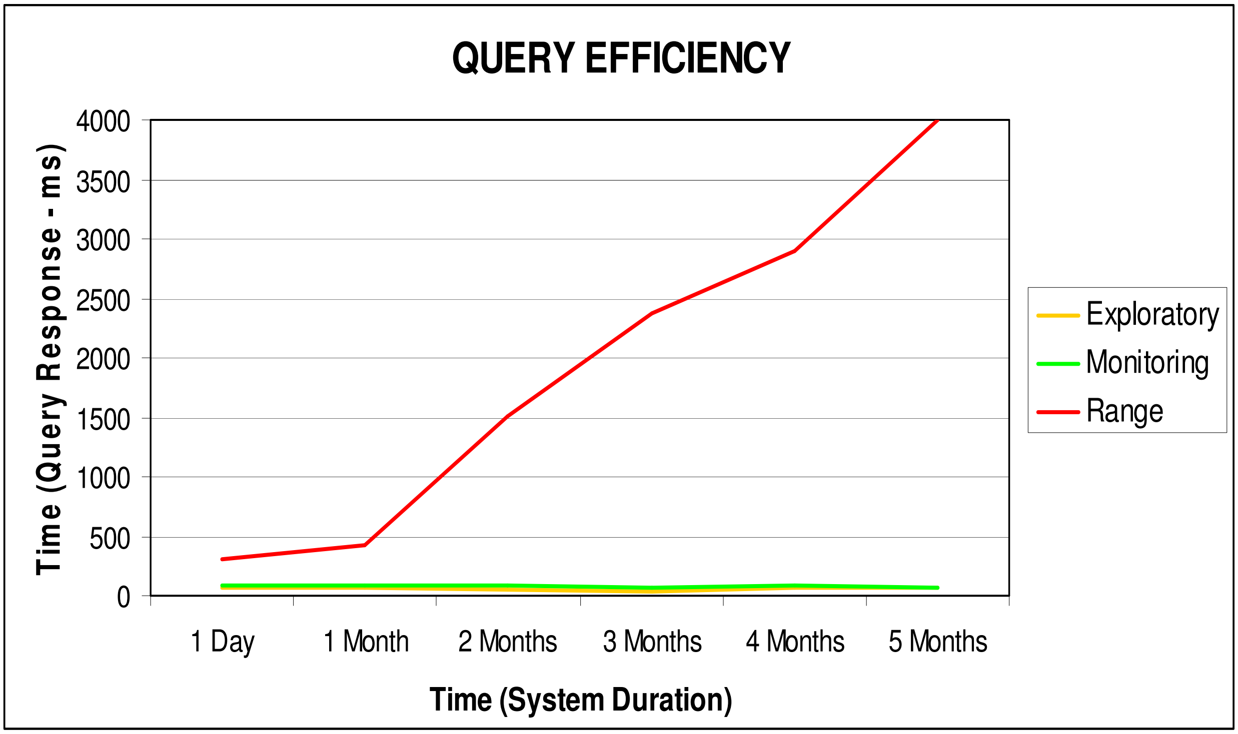


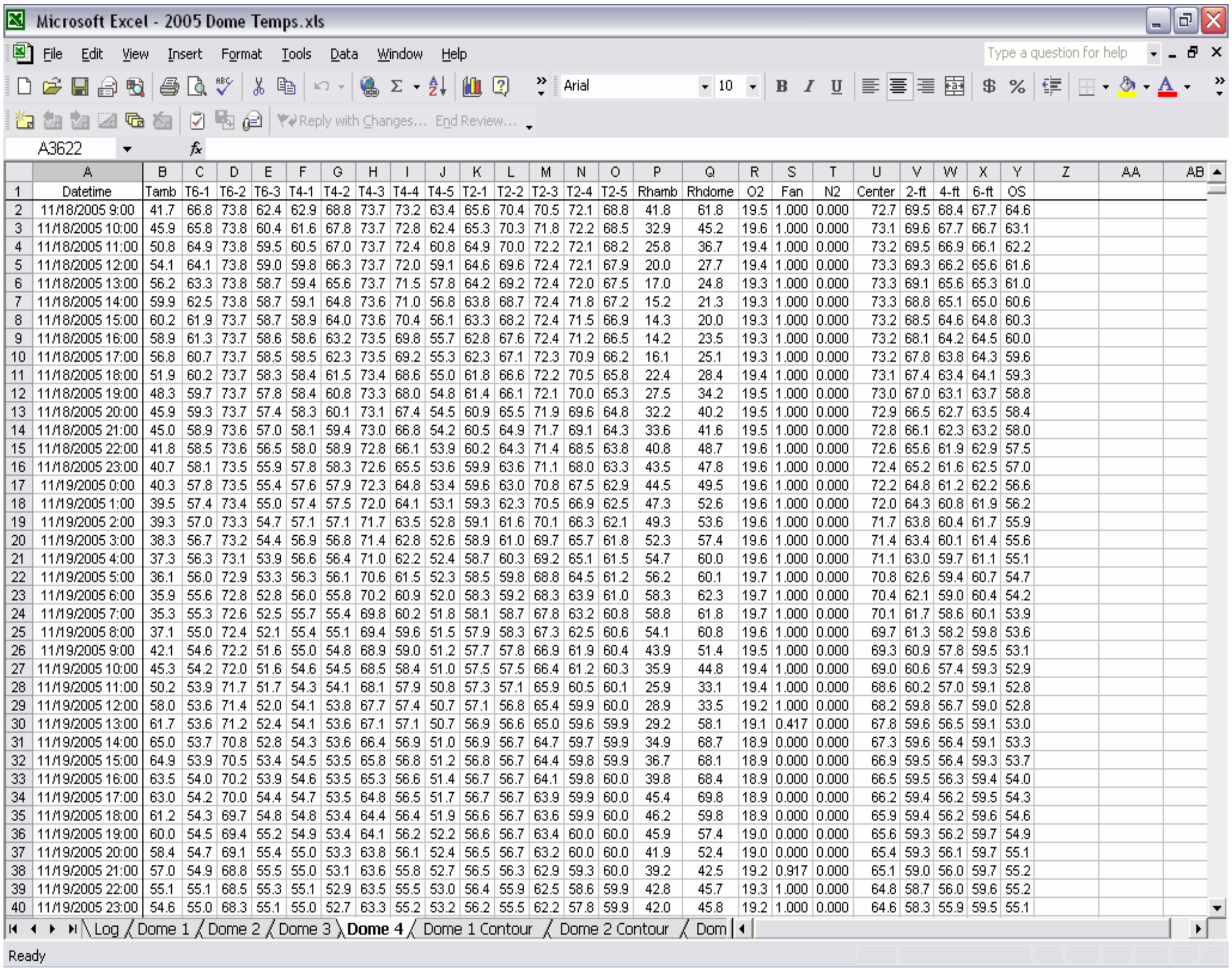
Figure 5.3: Query Efficiency Evaluation

In the figure above, the performance of the system is illustrated as it refers to query response time. The response times for the exploratory and monitoring queries (refer to chapter four, section five) fluctuate slightly but do not increase with time duration of the system. However, the response time for range queries increases almost linearly to the increase in time duration of the system. This is because at each monthly period more files have to be retrieved to satisfy the query.

Table 5.1: Query Response Evaluation Table

	Query Response Time (milliseconds)			Ontology Size (KB)
	Exploratory Query	Monitoring Query	Range Query	
1 Day	60	80	310	20
1 Month	60	90	431	600
2 Months	50	80	1502	1220
3 Months	40	70	2373	1840
4 Months	60	80	2905	2400
5 Months	70	70	3998	3020

The last two figures show how our system improved upon the data storage and management of that present at the USDA. Figure 5.4 is a snapshot of the Microsoft Excel spreadsheet capturing the data for November 19, 2005. Figure 5.5 shows the query page in our GUI after a query was posted for data on November 19, 2005.



	Datetime	Tamb	T6-1	T6-2	T6-3	T4-1	T4-2	T4-3	T4-4	T4-5	T2-1	T2-2	T2-3	T2-4	T2-5	Rhamb	Rhdome	O2	Fan	N2	Center	2-ft	4-ft	6-ft	OS
2	11/18/2005 9:00	41.7	66.8	73.8	62.4	62.9	68.8	73.7	73.2	63.4	65.6	70.4	70.5	72.1	68.8	41.8	61.8	19.5	1.000	0.000	72.7	69.5	68.4	67.7	64.6
3	11/18/2005 10:00	45.9	65.8	73.8	60.4	61.6	67.8	73.7	72.8	62.4	65.3	70.3	71.8	72.2	68.5	32.9	45.2	19.6	1.000	0.000	73.1	69.6	67.7	66.7	63.1
4	11/18/2005 11:00	50.8	64.9	73.8	59.5	60.5	67.0	73.7	72.4	60.8	64.9	70.0	72.2	72.1	68.2	25.8	36.7	19.4	1.000	0.000	73.2	69.5	66.9	66.1	62.2
5	11/18/2005 12:00	54.1	64.1	73.8	59.0	59.8	66.3	73.7	72.0	59.1	64.6	69.6	72.4	72.1	67.9	20.0	27.7	19.4	1.000	0.000	73.3	69.3	66.2	65.6	61.6
6	11/18/2005 13:00	56.2	63.3	73.8	58.7	59.4	65.6	73.7	71.5	57.8	64.2	69.2	72.4	72.0	67.5	17.0	24.8	19.3	1.000	0.000	73.3	69.1	65.6	65.3	61.0
7	11/18/2005 14:00	59.9	62.5	73.8	58.7	59.1	64.8	73.6	71.0	56.8	63.8	68.7	72.4	71.8	67.2	15.2	21.3	19.3	1.000	0.000	73.3	68.8	65.1	65.0	60.6
8	11/18/2005 15:00	60.2	61.9	73.7	58.7	58.9	64.0	73.6	70.4	56.1	63.3	68.2	72.4	71.5	66.9	14.3	20.0	19.3	1.000	0.000	73.2	68.5	64.6	64.8	60.3
9	11/18/2005 16:00	58.9	61.3	73.7	58.6	58.6	63.2	73.5	69.8	55.7	62.8	67.6	72.4	71.2	66.5	14.2	23.5	19.3	1.000	0.000	73.2	68.1	64.2	64.5	60.0
10	11/18/2005 17:00	56.8	60.7	73.7	58.5	58.5	62.3	73.5	69.2	55.3	62.3	67.1	72.3	70.9	66.2	16.1	25.1	19.3	1.000	0.000	73.2	67.8	63.8	64.3	59.6
11	11/18/2005 18:00	51.9	60.2	73.7	58.3	58.4	61.5	73.4	68.6	55.0	61.8	66.6	72.2	70.5	65.8	22.4	28.4	19.4	1.000	0.000	73.1	67.4	63.4	64.1	59.3
12	11/18/2005 19:00	48.3	59.7	73.7	57.8	58.4	60.8	73.3	68.0	54.8	61.4	66.1	72.1	70.0	65.3	27.5	34.2	19.5	1.000	0.000	73.0	67.0	63.1	63.7	58.8
13	11/18/2005 20:00	45.9	59.3	73.7	57.4	58.3	60.1	73.1	67.4	54.5	60.9	65.5	71.9	69.6	64.8	32.2	40.2	19.5	1.000	0.000	72.9	66.5	62.7	63.5	58.4
14	11/18/2005 21:00	45.0	58.9	73.6	57.0	58.1	59.4	73.0	66.8	54.2	60.5	64.9	71.7	69.1	64.3	33.6	41.6	19.5	1.000	0.000	72.8	66.1	62.3	63.2	58.0
15	11/18/2005 22:00	41.8	58.5	73.6	56.5	58.0	58.9	72.8	66.1	53.9	60.2	64.3	71.4	68.5	63.8	40.8	48.7	19.6	1.000	0.000	72.6	65.6	61.9	62.9	57.5
16	11/18/2005 23:00	40.7	58.1	73.5	55.9	57.8	58.3	72.6	65.5	53.6	59.9	63.6	71.1	68.0	63.3	43.5	47.8	19.6	1.000	0.000	72.4	65.2	61.6	62.5	57.0
17	11/19/2005 0:00	40.3	57.8	73.5	55.4	57.6	57.9	72.3	64.8	53.4	59.6	63.0	70.8	67.5	62.9	44.5	49.5	19.6	1.000	0.000	72.2	64.8	61.2	62.2	56.6
18	11/19/2005 1:00	39.5	57.4	73.4	55.0	57.4	57.5	72.0	64.1	53.1	59.3	62.3	70.5	66.9	62.5	47.3	52.6	19.6	1.000	0.000	72.0	64.3	60.8	61.9	56.2
19	11/19/2005 2:00	39.3	57.0	73.3	54.7	57.1	57.1	71.7	63.5	52.8	59.1	61.6	70.1	66.3	62.1	49.3	53.6	19.6	1.000	0.000	71.7	63.8	60.4	61.7	55.9
20	11/19/2005 3:00	38.3	56.7	73.2	54.4	56.9	56.8	71.4	62.8	52.6	58.9	61.0	69.7	65.7	61.8	52.3	57.4	19.6	1.000	0.000	71.4	63.4	60.1	61.4	55.6
21	11/19/2005 4:00	37.3	56.3	73.1	53.9	56.6	56.4	71.0	62.2	52.4	58.7	60.3	69.2	65.1	61.5	54.7	60.0	19.6	1.000	0.000	71.1	63.0	59.7	61.1	55.1
22	11/19/2005 5:00	36.1	56.0	72.9	53.3	56.3	56.1	70.6	61.5	52.3	58.5	59.8	68.8	64.5	61.2	56.2	60.1	19.7	1.000	0.000	70.8	62.6	59.4	60.7	54.7
23	11/19/2005 6:00	35.9	55.6	72.8	52.8	56.0	55.8	70.2	60.9	52.0	58.3	59.2	68.3	63.9	61.0	58.3	62.3	19.7	1.000	0.000	70.4	62.1	59.0	60.4	54.2
24	11/19/2005 7:00	35.3	55.3	72.6	52.5	55.7	55.4	69.8	60.2	51.8	58.1	58.7	67.8	63.2	60.8	58.8	61.8	19.7	1.000	0.000	70.1	61.7	58.6	60.1	53.9
25	11/19/2005 8:00	37.1	55.0	72.4	52.1	55.4	55.1	69.4	59.6	51.5	57.9	58.3	67.3	62.5	60.6	54.1	60.8	19.6	1.000	0.000	69.7	61.3	58.2	59.8	53.6
26	11/19/2005 9:00	42.1	54.6	72.2	51.6	55.0	54.8	68.9	59.0	51.2	57.7	57.8	66.9	61.9	60.4	43.9	51.4	19.5	1.000	0.000	69.3	60.9	57.8	59.5	53.1
27	11/19/2005 10:00	45.3	54.2	72.0	51.6	54.6	54.5	68.5	58.4	51.0	57.5	57.5	66.4	61.2	60.3	35.9	44.8	19.4	1.000	0.000	69.0	60.6	57.4	59.3	52.9
28	11/19/2005 11:00	50.2	53.9	71.7	51.7	54.3	54.1	68.1	57.9	50.8	57.3	57.1	65.9	60.5	60.1	25.9	33.1	19.4	1.000	0.000	68.6	60.2	57.0	59.1	52.8
29	11/19/2005 12:00	58.0	53.6	71.4	52.0	54.1	53.8	67.7	57.4	50.7	57.1	56.8	65.4	59.9	60.0	28.9	33.5	19.2	1.000	0.000	68.2	59.8	56.7	59.0	52.8
30	11/19/2005 13:00	61.7	53.6	71.2	52.4	54.1	53.6	67.1	57.1	50.7	56.9	56.6	65.0	59.6	59.9	29.2	58.1	19.1	0.417	0.000	67.8	59.6	56.5	59.1	53.0
31	11/19/2005 14:00	65.0	53.7	70.8	52.8	54.3	53.6	66.4	56.9	51.0	56.9	56.7	64.7	59.7	59.9	34.9	68.7	18.9	0.000	0.000	67.3	59.6	56.4	59.1	53.3
32	11/19/2005 15:00	64.9	53.9	70.5	53.4	54.5	53.5	65.8	56.8	51.2	56.8	56.7	64.4	59.8	59.9	36.7	68.1	18.9	0.000	0.000	66.9	59.5	56.4	59.3	53.7
33	11/19/2005 16:00	63.5	54.0	70.2	53.9	54.6	53.5	65.3	56.6	51.4	56.7	56.7	64.1	59.8	60.0	39.8	68.4	18.9	0.000	0.000	66.5	59.5	56.3	59.4	54.0
34	11/19/2005 17:00	63.0	54.2	70.0	54.4	54.7	53.5	64.8	56.5	51.7	56.7	56.7	63.9	59.9	60.0	45.4	69.8	18.9	0.000	0.000	66.2	59.4	56.2	59.5	54.3
35	11/19/2005 18:00	61.2	54.3	69.7	54.8	54.8	53.4	64.4	56.4	51.9	56.6	56.7	63.6	59.9	60.0	46.2	59.8	18.9	0.000	0.000	65.9	59.4	56.2	59.6	54.6
36	11/19/2005 19:00	60.0	54.5	69.4	55.2	54.9	53.4	64.1	56.2	52.2	56.6	56.7	63.4	60.0	60.0	45.9	57.4	19.0	0.000	0.000	65.6	59.3	56.2	59.7	54.9
37	11/19/2005 20:00	58.4	54.7	69.1	55.4	55.0	53.3	63.8	56.1	52.4	56.5	56.7	63.2	60.0	60.0	41.9	52.4	19.0	0.000	0.000	65.4	59.3	56.1	59.7	55.1
38	11/19/2005 21:00	57.0	54.9	68.8	55.5	55.0	53.1	63.6	55.8	52.7	56.5	56.3	62.9	59.3	60.0	39.2	42.5	19.2	0.917	0.000	65.1	59.0	56.0	59.7	55.2
39	11/19/2005 22:00	55.1	55.1	68.5	55.3	55.1	52.9	63.5	55.5	53.0	56.4	55.9	62.5	58.6	59.9	42.8	45.7	19.3	1.000	0.000	64.8	58.7	56.0	59.6	55.2
40	11/19/2005 23:00	54.6	55.0	68.3	55.1	55.0	52.7	63.3	55.2	53.2	56.2	55.5	62.2	57.8	59.9	42.0	45.8	19.2	1.000	0.000	64.6	58.3	55.9	59.5	55.1

Figure 5.4: Excel Snapshot of Data from 11/19/05

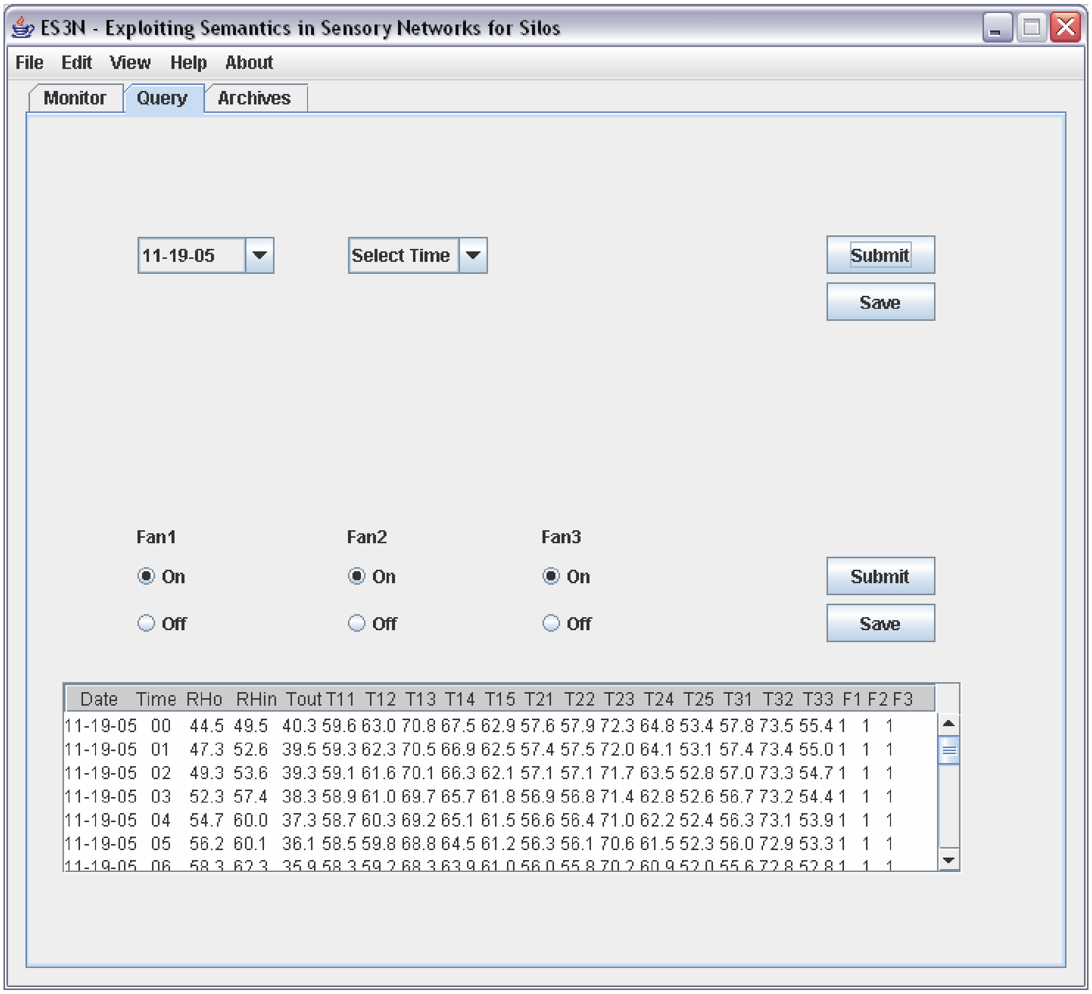


Figure 5.5: Snapshot of Query of Data from 11/19/05

At the conclusion of testing, we were pleased with the performance of our system. All proposed functionality was evaluated and approved in terms of the correlation to the system present at the USDA and the initial propositions made at the beginning of this thesis. The query tests showed concerns and areas that we should strengthen to support running the system for a long period of time.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

The primary goal of this thesis was to propose a system to aid the agriculture industry in efficient storage of cereal grain and oil seed products. Test results show that the system is efficient, accurate, and user-friendly. Millions of dollars are lost each year from product spoilage during storage; and with the strict standards and guidelines enforced by the USDA and the ASAE to diminish health hazards, farmers and anyone storing such products have to update antiquated storage techniques.

The random sampling was replaced with the implementation of a sensor network as a mechanism for data acquisition, creating an automated monitoring system. This provides consistency and eliminates the need for human interaction in data retrieval. The absence of data storage was corrected with the use of a dynamic ontology. This was the first exhibition of Semantic Web techniques in our system. The data management issue was rectified by presenting the data semantically, such that relationships and associations among the data could be uncovered and used for querying.

Another goal of this thesis was to expose how concepts and knowledge gained here in the Computer Science Department at the University of Georgia could be applied to solve a real-life problem. Software development, real-time monitoring, data storage and management, Semantic Web; these are all skills gained or at least cultivated here at UGA. Gaining such knowledge is important, but being able to apply that knowledge is of greater value.

The system in its present stage functions adequately; however, there are a few future additions to be made before implementation at Cargill and/or the USDA. The biggest enhancement is the addition of an RDF repository such as BRAHMS (Janik, 2005) or Sesame (Sesame, 2006). Both are RDF stores capable of storing extremely large amounts of RDF data. They both also support fast semantic association discovery in large RDF bases. With hourly readings, the system deals with over 500 triples daily. It accumulates 7.3 MB of data yearly. Without the addition of an RDF store, the system would suffer a lack in query efficiency after being implemented for several years.

Another addition includes more sensors to aid in monitoring and further diversify the system. In chapter one, we discussed research that had been done comparing the effectiveness of a CO_2 sensor with that of the traditional temperature sensor (Ileleji, 2006). Studies showed that carbon dioxide detection is more prone to catching product deterioration in its earlier stages than the detection of a hot spot within the stored product. The cost of CO_2 sensors is coming down; therefore, adding one to this system would be feasible. Since there was no real data available to support it, no CO_2 sensor was used in this thesis. The other useful addition is a sonar sensor to be used in level detection of the stored product within the silo. The sonar would be located at the top of the silo and send impulses down to the stored product. Depending on the time it would take for the sonar to receive these impulses, a calculation could be made to determine the height of the stored product within the silo. This feature would also make the user aware of the silo being loaded or unloaded. These additions would require no change in the system's hardware. The two sensors would merely be additions to the already existing sensor network.

The last enhancement, that of artificial intelligence (AI), would make the system energy efficient in that it would ensure that the fans run no more than needed. AI would benefit our system through the addition of a fuzzy logic controller to operate the fans. Fuzzy logic is a multivalued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, on/off, hot/cold or high/low (Hellmann, 2001). With a controller as such, thresholds could be set to optimize the operation of fans, minimizing the time the fans are on but at the same time maximizing the benefit of aeration to the stored product. However, to experience the full benefit of a fuzzy logic controller, the fans would need to have the option of variable speeds. The present fans discussed in this thesis only have one speed. Therefore, this last enhancement would require a change in the system's hardware in order for it to be implemented; but we are confident that the alteration would be well worth the trouble in the long run.

In the upcoming months, communication will be done with Cargill and USDA as we head toward the implementation of our system. Work will be done with both parties as the system is conformed to an application to be presented to users in the agricultural industry.

BIBLIOGRAPHY

Antoniou, G., and Harmelen, F. van. (2004). *A Semantic Web primer*. Cambridge, Mass: The MIT Press.

ASAE. (2002). *ASAE Standards 2002: 49th Edition, Standards Engineering Practices Data*. American Society of Agricultural Engineers.

Beedle, P. L. (2001). *Silos: An agricultural success story*. Madison, WI: University of Wisconsin-Extension.

Bokor, Gabe. (2000). "Search Engines Revisited". *Translation Journal*, Vol 4(4). Retrieved November 18, 2006 from http://www.accurapid.com/journal/14search.htm

Brickley, D., and Guha, R. (2000). *Resource Description Framework (RDF) Schema Specification 1.0*. Retrieved November 17, 2006 from http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

Broehl Jr., W.G. (1992). *Cargill: Trading the world's grain*. Hanover, NH: University Press of New England.

Broehl Jr., W.G. (1998). *Cargill: going global*. Hanover, NH: University Press of New England.

Butts, C.L., Arthur, F.H., Brown, S.L., and Dorner, J.W. (2006). "Aerating farmer stock peanut storage in the southeastern U.S." *Transactions of the American Society of Agricultural and Biological Engineers.* Vol 49(2), p. 457-465.

Canadian Grain Commission. (2004). *Managing the quality of stored grain: controlling insect infestations*. Retrieved October 31, 2006 from http://www.grainscanada.gc.ca/Entomology/control01-e.htm#fumigation

Cargill. (2006). *Cargill Industries Inc*. Retrieved May 1, 2006 from http://www.cargill.com

Charny, B. (2002, December). Wireless research senses the future. *ZDNet News*. Retrieved November 4, 2006 from http://news.zdnet.com/2100-1009_22-976377.html

Childs, J. (2006, October). Camvera Partners with Organizations Across Georgia to Make Rural Broadband a Reality. *CCN Matthews Co. Market Wire News*. Retrieved November 19, 2006 from http://www.marketwire.com/mw/release_html_b1?release_id=171844

Chong, Chee-Yee, and Kumar, S.P. (2003). "Sensor networks: evolution, opportunities, and challenges". Proceedings of the Institute of Electrical and Electronics Engineers. Vol 91(8), p.1247-1256.

DAML. (2006). *The DARPA Agent Markup Language homepage*. Retrieved November 16, 2006 from http://www.daml.org/

Decker S., Melnik S., Harmelen F., Fensel D., Klein M., Broekstra J., Erdmann M., and Horrocks, I. (2000). "The Semantic Web: The roles of XML and RDF". *Institute of Electrical and Electronics Engineers Internet Computing*. Vol. 4(5), p. 63-74.

European Commission. (2002). *The sixth framework programme*. Retrieved November 19, 2006 from http://ec.europa.eu/research/fp6/pdf/fp6-in-brief_en.pdf

Fensel, D., Wahlster, W., Lieberman, H., and Hendler, J. (2002). *Spinning the Semantic Web: bringing the World Wide Web to its full potential.* Cambridge, Mass: The MIT Press.

Gould, L.S. (2005, June). If only drivers were as attentive as sensor automation. *Automotive Design & Production*. Retrieved October 31, 2006 from http://www.autofieldguide.com/articles/060507.html

Gruninger, M., and Fox, M. (1995). "Methodology for the design and evaluation of ontologies". *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing in the International Joint Conference on Artificial Intelligence*. Montreal, Canada, p.1-10.

Haenselmann, T. (2006) *Sensor Networks*. Retrieved October 30, 2006 from http://www.informatik.uni-mannheim.de/~haensel/sn_book/sensor_networks.pdf

Halpin, H. (2004). *The Semantic Web: The origins of artificial intelligence redux*. Scotland, UK: University of Edinburgh.

Hayes, P. (2004). *RDF Semantics: technical report, W3C*. Retrieved November 17, 2006 from http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Heflin, J. (2004). *OWL Web Ontology Language use cases and requirements*. Retrieved November 17, 2006 from http://www.w3.org/TR/2004/REC-webont-req-20040210/

Hellmann, M. (2001). *Fuzzy Logic Introduction*. Retrieved November 18, 2006 from http://www.fpk.tu-berlin.de/~anderl/epsilon/fuzzyintro4.pdf

Hills, R. (2001, July/August). Sensing for danger. *Science and Technology Review*. Retrieved October 30, 2006 from http://www.llnl.gov/str/JulAug01/Hills.html

Ileleji, K.E., Maier, D.E., Bhat, C., and Woloshuk, C.P. (2006). "Detection of a developing hot spot in stored corn with a CO2 sensor". *Transactions of the American Society of Agricultural Engineers*. Vol. 22(2), p. 275-289.

Janik, M., and Kochut, K. (2005). "BRAHMS: A workbench RDF store and high performance memory system for semantic association discovery". *The Fourth International Semantic Web Conference*. Galway, Ireland, p.431-445.

Jensen, D. (2002, June), SIVAM: Communication, navigation and surveillance for the Amazon. *Avionics Magazine*. Retrieved October 30, 2006 from http://www.aviationtoday.com/reports/avionics/previous/0602/0602sivam.htm

Juvva, K. (1998). *Real-Time Systems*. Retrieved November 5, 2006 from http://www.ece.cmu.edu/~koopman/des_s99/real_time/

Kumar, S., and Shepherd, D. (2001). "SensIT: Sensor information technology for the warfighter". *Proceedings of the 4th International Conference on Information Fusion*. Montreal, Canada, pp. TuC1-3 – TuC1-9.

Levinson, H., and Levinson, A. (1994). "Origin of grain storage and insect species consuming desiccated food". *Journal for Pest Science: Biomedical and Life Sciences*. Vol 67(3), p. 47 – 60.

Lewis, M., Cameron, D., Xie, S., and Arpinar, I.B. (2006). "ES3N: A semantic approach to data management in sensor networks". *Semantic Sensor Network Workshop in the Fifth International Semantic Web Conference*. Athens, GA, p. 1-13.

Liu, S. (2000). Chapter 2: Hard versus soft real-time systems. In *Real-time systems* (p. 26-33). New Jersey: Prentice Hall.

Monitor. (2006). *Monitor Technologies LLC*. Retrieved October 30, 2006 from http://www.monitortech.com/

Noy, N. and McGuinness, D. (2001). "Ontology Development 101: a guide to creating your first ontology". *Stanford Knowledge Systems Laboratory Technical Report*. KSL-01-05, p. 1-25.

Parker, C., Shields, J., Armitage, D., Hook, S., Knight, J., McConnell, S., and Williams, R. (2006). "Grain store management support using the GrainPlan DSS". *Proceedings of the 4th World Congress Conference on Computers in Agriculture and Natural Resources*. Orlando, FL, p. 463-468.

Rukunudin, I.H., Bern, C.J., Misra, M.K., and Bailey, T.B. (2004). "Carbon dioxide evolution from fresh and preserved soybeans". *Transactions of the American Society of Agricultural Engineers.* Vol 47(3), p. 827 – 833.

Sesame. (2006). …*Home of Sesame*. Retrieved from November 19, 2006 http://www.openrdf.org/

Shadbolt, N., Berners-Lee, T., and. Hall, W. (2006). "The Semantic Web revisited". *Institute of Electrical and Electronics Engineers Intelligent Systems*. Vol 21(3), p. 96-101.

Shuman, D., Larson, R.G., and Epsky, N. (2004). "A quantitative stored-product insect monitoring system using sensor output analog processing (SOAP)". *Transactions of the American Society of Agricultural Engineers.* Vol 47(5), p.1857-1864.

Shuman, D., and Epsky, N. (1999). "Computerized monitoring of stored-product insect population". Proceedings of the 7th International Working Conference on Stored-Product Protection. Chengdu, China, p. 1429 – 1436.

SiloPatrol. (2004). *SiloTrack™ Inventory Management Software Version 3.0*. Bulletin 343B. Retrieved October 30, 2006 from http://www.monitortech.com/pdfs/bulletins/SPSiloTrack343B.pdf

Sinha, R.N., and Wallace, H. (1965). "Ecology of a fungus-induced hot spot in stored grain". *Canadian Journal of Plant Science.* Vol 45(1), p. 48-59.

Staab, S., and Studer, R. (2004). *Handbook on ontologies*. New York: Springer.

Steer, D., Baptista, A., McNamee, D., Pu, C., and Walpole, J. (2000). "Research challenges in environmental observation and forecasting systems". *Proceedings of the 6th International Conference on Mobile Computing and Networking (MOBICOMM).* Boston, MA, pp. 292-299.

Stewart, M. (2001). *Tim Berners-Lee: Inventor of the World Wide Web*. New York: Ferguson Publishing Company.

Uschold, M., and Gruninger, M. (1996). "Ontologies: principles, methods and applications". *Knowledge Engineering Review*. Vol 11(2), p. 93-155.

Walton, C. (2006). *Agency and the Semantic Web*. New York: Oxford University Press.

Wilcke, B. (1998). *Converting tower silos for dry grain storage*. Twin Cities, MN: University of Minnesota Extension Service.

Williams, J.H., Phillips, T.D., Jolly, P.E., Stiles, J.K., Jolly, C.M., and Aggarwal, D. (2004) "Human aflatoxicosis in developing countries: a review of toxicology, exposure, potential health consequences, and interventions". *American Journal of Clinical Nutrition*. Vol 80(5), p. 1106-1122.

APPENDIX A

ONTOLOGY SCHEMA FOR DATA INSTANCES

The following section of OWL and RDFS shows the schema for the ontology. The properties are in order as they appear in the ontology and in results when a query is posted. These properties correlate to what would be identified as attributes in a relational database. The properties listed below are "has_date", "has_time", "has_humidityout", "has_humidityin", "has_tempout", "has_temp11", "has_temp12", "has_temp13", "has_temp14", "has_temp15", "has_temp21", "has_temp22", "has_temp23", "has_temp24", "has_temp25", "has_temp31", "has_temp32", "has_temp33", "has_fan1", "has_fan2", and "has_fan3". These properties are explained in chapter four, section four. The values for the properties are defined as strings within the schema; however, they are cast to their appropriate data type within the system.

```
<owl:DatatypeProperty rdf:ID="has_date">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
   <rdfs:domain rdf:resource="#Data"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="has_time">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
   <rdfs:domain rdf:resource="#Data"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="has_humidityout">
   <rdfs:domain rdf:resource="#Data"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="has_humidityin">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
   <rdfs:domain rdf:resource="#Data"/>
 </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_tempout">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
   <rdfs:domain rdf:resource="#Data"/>
```

```
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp11">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp12">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp13">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="has_temp14">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp15">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp21">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp22">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp23">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp24">
    <rdfs:domain rdf:resource="#Data"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_temp25">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp31">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp32">
```

```
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_temp33">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_fan1">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="has_fan2">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="has_fan3">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain rdf:resource="#Data"/>
  </owl:DatatypeProperty>
```