8-8-2009

Standards-based sensor web for wide area monitoring of power systems

Nischal Dahal

STANDARDS-BASED SENSOR WEB FOR WIDE AREA MONITORING OF POWER SYSTEMS

By

Nischal Dahal

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of Requirements
for the Degree of Master of Science
in Computer Engineering
in the Department of Electrical And Computer Engineering

Mississippi State, Mississippi

August 2009

STANDARDS-BASED SENSOR WEB FOR WIDE AREA

MONITORING OF POWER SYSTEMS

By

Nischal Dahal

Approved

Nicolas H. Younan
Professor and Interim Department Head
Department of Electrical and Computer
Engineering (Major Advisor)

Surya S. Durbha
Assistant Research Professor,
Geosystems Research Institute (GRI)
(Co-Major Advisor)

Andy D. Perkins
Assistant Professor
Department of Computer Science and
Engineering (Committee Member)

James E. Fowler
Professor and Interim Graduate Program
Director
Department of Electrical and Computer
Engineering

Sarah A. Rajala
Dean, Bagley College of Engineering

Name: Nischal Dahal

Date of Degree: August 8^{th}, 2009

Institution: Mississippi State University

Major Field: Computer Engineering

Major Advisor: Dr. Nicolas H. Younan

Title of Study: STANDARDS-BASED SENSOR WEB FOR WIDE AREA MONITORING OF POWER SYSTEMS

Pages in Study: 68

Candidate for Degree of Master of Science

The balance of supply and demand of energy is the key factor in the stability of power systems. A small disturbance in the supply demand relationship, if not properly handled, can cascade into a major outage, costing millions of dollars. Proper monitoring and real-time information exchange is the only prevention measure in this vulnerable system. But, the disparity in the protocols used by power utilities and the lack of infrastructure for information exchange are proving to be hindrance towards a reliable de-regularized power industry. In this thesis, an emerging SWE has been adapted for the wide area monitoring of power systems. SWE and CIM provide a solution to both problems of heterogeneity and lack of central repository of the data. The sensor data from utilities, published in CIM, are exposed via a SOS. This provides a standard method for discovering and accessing the sensor data between utilities facilitating rapid response.

ACKNOWLEDGEMENTS

I would like to thank my major advisor Dr. Nicolas H. Younan for his continued trust, support, and guidance from the beginning of this research. I would like to thank my co-major advisor Dr. Surya S. Durbha for his time and guidance while working on this research. I would like to acknowledge my committee member, Dr. Andy D. Perkins, for his valuable comments and suggestions in preparation of this thesis.

I would like to show my gratitude towards the Department of Electrical And Computer Engineering, for providing the financial assistance and facilities, which made this work possible. I would like to thank fellow graduate students and faculties of the Department of Electrical and Computer Engineering who worked in this project for their support and advices. I would like to thank the Oak Ridge National Laboratory (ORNL) and the Department of Homeland Security (DHS) for funding this project.

Finally, I would like to thank my family for their unconditional and continuous love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background

The balance between the demand and supply of power systems is one of the most important factors for stability. A small disturbance in the demand supply balance can cascade into an outrageous blackout over a large region, depleting millions of dollars from stakeholders. In order to ensure continuous operation, the stability of the electrical system has to be constantly monitored for any inconsistency.

The power industry is highly interrelated with each other. The propagation of a disturbance can only be prevented if the utility companies are also aware of the developing conditions in neighboring utilities, in addition to their own system [1]. Effective communication between utility companies can be critical in ensuring the seamless operation of the whole system. The recent de-regularization of the power industry has increased the need of mutual information sharing among the utilities.

Most of the communication protocols being used in power systems are either vendor proprietary or user developed standards, which are incompatible with each other. This incompatibility increases the cost of information exchange between utilities[1].The lack of standardization in the data exchange mechanism and the information monitoring mechanism are impediments for effective decision making. Proper information exchange at the appropriate time is essential for the prevention of outages. In recent history, numerous outages can be found which could have been prevented with proper exchange of information among utilities [1].

Because of the billions of dollars already invested in the existing infrastructure and training of staff, it is very important that the new system be compatible with existing systems. The cost of a new system to be built should be kept as low as possible while utilizing the maximum possible part of the existing system.

On the other hand, the current system does not provide leverage to security coordinators [2] of a region to monitor all the systems within their region with a single application. The precious time, which can be used to act upon the contingency, is wasted just to remove the heterogeneities and acquire the relevant information. A central information repository and query system can empower the security coordinator for coordinated actions over a wide region to prevent contingencies.

Sensor Web Enablement Approach for Wide Area Monitoring

Sensor Web Enablement is a protocol for innovative acquisition of sensor data implemented over the World Wide Web (WWW) for fully interoperable information exchange between all sensor systems. The seamless access to real time or near real time power systems sensors data is constrained by varying characteristics (physical/logical) of the sensor networks. Sensor Web Enablement (SWE) [3] provides a solution to the problems of heterogeneity of data and lack of central repository of data for proper action in case of contingency. Sensor Web is an emerging technology trend towards achieving a collaborative, coherent, consistent, and consolidated method of sensor data collection, fusion, and distribution. It adds a sensor dimension to the Internet that allows users to glean meaningful information about the sensor observation via a web browser. As illustrated in Figure 1, SWE acts as a common interface to data from electrical utilities for the security coordinator, thus facilitating the seamless access to data and reconciling the heterogeneities in various interdependent systems.

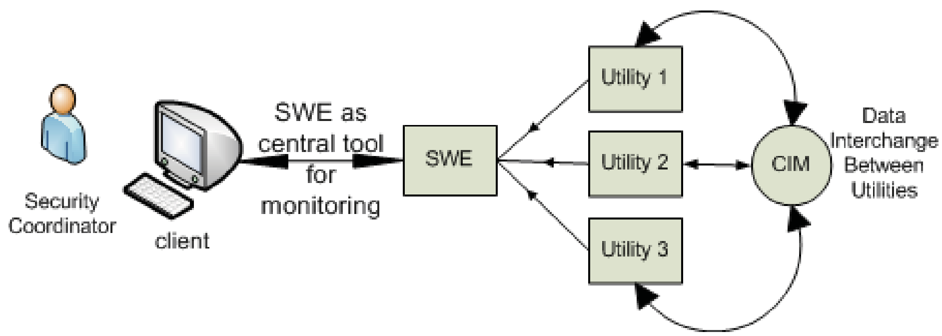

Figure 1 Communication of Security Coordinator with utilities within a region

The security coordinator can access the data of all utilities in his/her region from a single client application to decide on the corrective measures for the contingency. In addition to adding value to monitoring his/her own region, the SWE also enables free communication among security coordinators of the neighboring regions, as shown in Figure 2. The effective communication among security coordinators is vital for preventing contingencies. Currently, the security coordinators are usually aware of the facts in their region but are not informed about the neighboring regions. Sensor Web provides an easy and cost effective legacy system that allows them to coordinate with neighboring utilities and neighboring security coordinators.

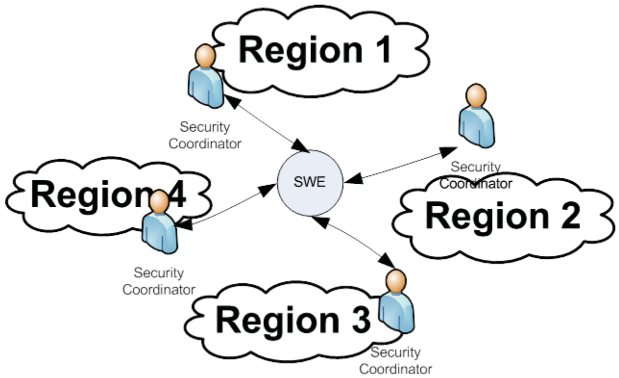

Figure 2 Communication among security coordinators of neighboring regions

Goal and Scope

In this research, we aim to build a standard-based wide area monitoring tool, which can be used to detect and counteract contingencies to prevent them from spreading over a wide area. Unlike traditional software currently being used in the power industry, the wide area monitoring tool will use global standards, such as Sensor Modeling Language (SensorML) [4], Sensor Observation Service (SOS) [5], and Common Information Model (CIM) [6] as the components of the application. The use of global standards will make it compatible with a wide range of applications. This compatibility of the newly developed tool will make the communication and integration with third party applications easy and cost effective.

The wide area monitoring application will act as a tool for security coordinators to monitor all the utilities within their region. The prototype will access a central repository of sensor data critical to the system stability, irrespective of the protocol used to acquire the data. The repository will contain sensor data from all the utilities in a region. The repository is exposed to the clients (primarily security coordinators) via Sensor

Observation Service (SOS), so that data can be accessed with easily available infrastructures, such as an internet connection and proper authentication.

The sensor data is published by the utility companies in Common Information Model (CIM) format. The CIM document is queried for the sensor data and stored in the central repository. As the sensors continuously monitor the electrical infrastructure, the sensor data also changes continuously. The utilities are expected to continuously publish the sensor data and keep its repository up-to-date. Figure 3 demonstrates an overview of the research.

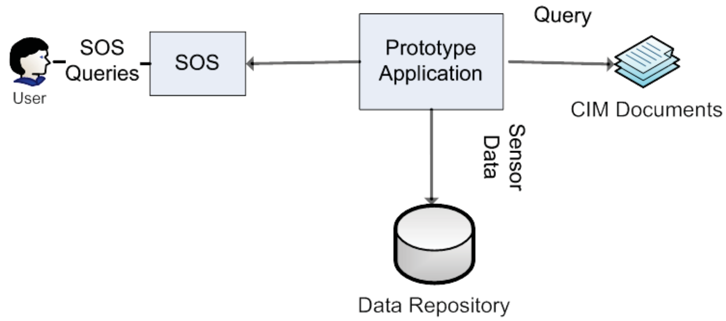

Figure 3 Overview of the Research Goal

The prototype will include an intelligent alert system, which will alert the users for any contingency detected based on the intelligence of the offline analysis of sensor data specific to the topology of the system, as shown in Figure 4. Each utility maintains a documentation of offline analysis, as its intelligence, for action plans to counter contingency. The intelligence of the offline analysis is intended to be digitized, so that the same tested intelligence can be retrieved quickly.

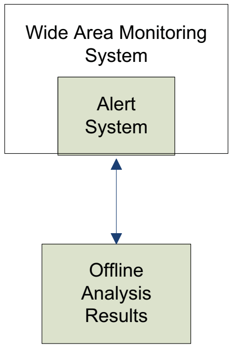

Figure 4 Alert System backed by offline analysis

Because of the complexity of the power system, The developed prototype is not expected to be used as an intelligent decision support system, rather act as an important decision making tool, which will be helpful for a quick access of accurate and meaningful information for prompt action to correct contingencies.

CHAPTER II

RELATED WORK

Background

Usually, power systems are designed with great care to minimize contingencies. However, because of the complexity of electrical systems, it is not possible to predict and prevent all types of contingencies [7]. Wide area monitoring of power systems helps early detection of a contingency, which is essential to prevent its spread over a large area. The growing electricity trading over sparse geographic areas is pushing the traditional monitoring system to its limits. Currently, the local remedial actions are mostly based on local information, which can lead to uncoordinated or even contradictory actions at different points of topology because the operators cannot perceive the global status of the system with only the local data [8]. In order to make actions taken at different point of topology coordinated, a centralized intelligence, which will help visualize the global status of the power systems, is necessary.

Wide area monitoring of the power grid consists of the following components:

- Data Acquisition
- Data Transport, Storage and Sharing
- Information Presentation
- Interoperability and Automation

The block diagram of the wide area monitoring system for power systems is shown in Figure 5. The data from sensors can be processed both at the substation level and the central level for proper information to be stored in the database.

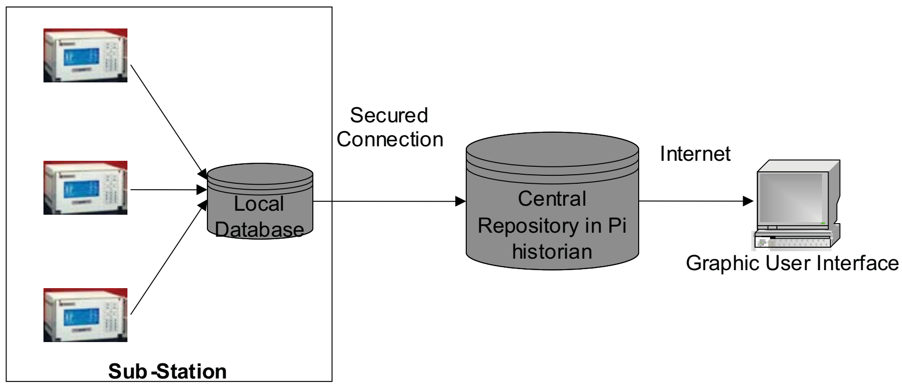

Figure 5 Block Diagram of the Wide Area Monitoring of Power Systems

Data Acquisition

The electrical system's conditions can change in seconds; the maintenance of any intelligence will do no good until it is kept abreast with the changing environment. The synchronization of measurements from different sensors over a wide area plays a key role in making the central intelligence reliable. The traditional sensors, such as line differential relays, need to communicate with each other for synchronization of the measurements. The increasing need of synchronization over a wide area is making traditional sensors obsolete [9].

The employment of new sensors, such as Phasor Measurement Unit (PMU), has made wide area monitoring more feasible than ever. Unlike the differential relays, the PMUs do not need to communicate with each other for synchronization. Instead, each measurement of PMU is auto-synchronized with a Global Positioning System (GPS) time [7]. The high speed of data acquisition of the PMUs with high accuracy makes the monitoring accurate regardless of the width of the geographic area under consideration. The increasing capacity and decreasing cost of communication and computation are making wide area monitoring more feasible than any time before [9].

Data transport, Storage, and Sharing

The medium of transportation of the data has to be resilient enough to ensure the delivery of data to the central repository even in extreme weather related events, such as tornados and hurricanes. With the affordability of high speed internet connection, a dedicated communication link using protocol TCP/IP [10], compatible with the internet, can be a very good option [7] [8].

The security of the communication channel is a major concern in the transportation of data from a substation to the central level. The critical information about the stability of the power system may be leaked to undesired elements, if appropriate measures of cyber security are not adopted. Updated encryption methods and proper monitoring measures have to be employed to ensure the security of the information being transferred [11] [12].

The PMUs are located in strategically important locations in power systems. All the PMUs are placed and monitored in a substation level with some redundancy to ensure accuracy [9] [13]. With a very high sample rate (10 to 80 samples/second), the infrastructure should be able to handle massive data. For example: the Phasor Data Concentrator (PDC) at Tennessee Valley Authority (TVA), which handles data from 90

PMUs, is handling 31 Gigabytes of data each day [14]. The storage of the data by the PDCs should be at the same rate as that of the data acquisition, while the consumer of the data may have different requirements [14]. Offline analysis applications may afford some delay in the data delivery, while real time monitoring applications cannot afford even few seconds of delay. On the other hand, the real time software requires data of small duration, while the offline data analysis software needs data of longer duration. So, there may be tradeoffs between parameters, such as the amount of data transported and time delay, in the requirement of data for a particular type of application [14]. Therefore, for maximum utilization of resources and time, the sharing of data by servers, such as Real-Time Dynamics Monitoring System (RTDMS) data management server, has to be customizable depending upon the type of application requesting data [14]. The RTDMS platform uses web services, such as Microsoft's Common Object Model (COM), for publishing the data in different specifications. The use of web services standards makes it easy to integrate with third party applications [14] [15]. In addition, web services can make the data available to diverse platforms, such as PDAs, PCs, laptops, clusters, etc [15].

Software, such as the OSIsoft's Pi historian, have to be used to collect and store sensor data for effective manipulation and processing of data [16]. Because of a very high phasor data acquisition rate, there may be multiple stages of data storage. The data may be concentrated locally for an area. The concentrated data can be locally processed for initial steps, such as time synchronization and removal of redundancy, and then forwarded to a central location for final processing and archiving [14]. Because of the lowering cost of high-end computational infrastructure, the pre-processing of the data can locally be realized without much economic burden.

Information Presentation

One of the most important features of any monitoring system is an alert feature. The prototyping applications on California ISO's phasor platform, Real-Time Dynamics Monitoring System (RTDMS), have a component called Real Time Alarm & Event Detector (RTAED). RTAED is responsible for processing real time information against a set of alarming criteria and event detection triggers [14]. In case of any event of contingency, the RTDMS server stores both pre-event and post event data, which can be useful for forensic analysis of the event [14]. The RTDMS also has options, such as data filtering, setting alarm, editing alarm, and event detection criteria. The alarm email server and report servers are responsible for sending out alert email notifications and generating the report, respectively. While on the client side, the RTDMS has different components, such as visualization and events, event analyzer, and report client. All the aforementioned components of the client can acquire information from the RTDMS server via different web services hosted by the server and display the results in graphical layouts with very simple computations [14].

The monitoring applications are generally designed to assist decision making in case of emergencies, so the response time of a user is critical. The identification of critical data for decision making is crucial for presenting the information without overwhelming the operator or security coordinator [14]. The information should be presented in a user-friendly Graphic User Interface (GUI) with the warning and alarm system in place to attract the attention of the user directly to the problem [17]. The GUI should be able to display data and trend yielding a snapshot of the power systems being monitored [9] [17]. The visualization of the data should be customizable. The user should be able to choose from 2-dimensional and 3-dimensional curves, pie chart and bar diagram, single phasor diagrams, single-line diagram, PV curve, etc [18].

Interoperability and Automation

A wide variety of software being used, with no formal standard of data representation within the application, has been impeding the communication between the applications [1]. With the government regulations slowly forcing the electric industry towards disassembling into transport, generation, and distribution sector, a common format of data exchange has become imminent [22].

Ontology is a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations [19]. The knowledge sharing with ontology will effectively share the knowledge among the software in the same domain in an unambiguous way, so that the information passed by the transmitter is correctly perceived by the receiver [20].

A fully machine readable and automatic reasoning system is needed for the artificial intelligence required for the revolutionary concepts, such as Automatic Operator [21]. Ontology perfectly fits into this concept. According to Studer [20], "Ontology is the explicit formalized norm explanation of the shared conceptualization model."

The Common Information Model (CIM) has been adopted by the electrical utility industry as a standard form for exchanging information among the utilities [1] [22] [23].

The CIM is developed by Electric Power Research Institute (EPRI) and has evolved to a global standard after being adopted by the International Electro-technical Commission (IEC) under the TC 57 framework. The CIM can be similar to ontology for the power industry, which can be expressed in universal formats, such as UML, XML, RDF, and OWL [22].

Ontology expressed in XML cannot support the important features of the ontology, such as knowledge interoperability, knowledge aggregation, and reasoning, because of the limitation of XML [24]. While moving towards a formal ontology, Resource Description Format (RDF) has the properties to support machine readable ontology. The RDF has a useful property of expressing the relationship between entities, which is used to encode knowledge in an XML document [23].

Although there are numerous advantages of the representation of knowledge in RDF, the RDF representation cannot express every kind of knowledge. For example, the knowledge with cardinalities restriction, such as a power sensor having at most one ID, cannot be expressed in RDF [24]. Web Ontology Language (OWL) has been especially designed by the World Wide Web Consortium (W3C) for more expressive ontology and machine interpretability [25]. CIM has also been released in OWL format [22].

Summary

With the increasing use of modern sensors, such as Phasor Measurement Units, wide area monitoring of power systems has become realizable for a much larger geographic areas. The increasing communication and computation capability with decreasing cost have extended the latest state-of-the-art technologies, even to microlevels in power industry. This effectively makes the power industry more reliable than ever realized before.

With the rapid development of information technologies, such as web services, information can now be customized according to the need of the user. A simple interface can hide the complex implementation details of the database from general users; thereby simplifying the access of data.

By using the latest trends and developments in the information technologies and modern sensors, the power industry, which has been conservative in exchanging information, is changing its face with standard-based knowledge sharing. With the evolution of concepts, such as automatic operators in the power industry, the need of universally understandable knowledge is being fulfilled with ontology. With the development of standards, such as OWL and RDF, the expressiveness of a wide range of knowledge through ontology is now possible.

The existing concept of the electrical industry, which depends upon human intelligence, would be well supported with PDCs acting as central intelligence. The operators rely on information from central archive of sensor data for coordinated actions. These actions of the operators are the most important steps towards the prevention of power system disasters, such as blackouts. The graphic user interface of the Wide Area Monitoring (WAM) system should give the operators a clear snapshot of the power system for quick and accurate decision making.

In this research, a WAM system based on different standards has been developed. The Open Geospatial Consortium (OGC) based standards, such as SensorML, Sensor Observation Service (SOS), and Observation and Measurement (OM), give the WAM compatibility with other sensor standards. For example, if some natural disasters occur, then the compatibility of power sensors standards with other sensors, such as wind speed sensors, can give the emergency management crew an extra hand in handling the emergency. Discovering and accessing sensor data eliminates the requirement of prior knowledge of a sensor in an area, which facilitates the rapid response towards normalizing the contingency.

This research also focuses towards the interoperability and communication among utilities. The use of Common Information Model (CIM) as a method of publishing sensor data from utility decouples the WAM from an individual utility. This approach is making more sense after the de-regularization of the electrical industry, which has forced the electrical industry to reconsider inter-utility communications.

CHAPTER III

SENSOR WEB ENABLEMENT

Introduction

There exist millions of power system sensors which have been employed to monitor numerous parameters, such as voltage, current, voltage angles etc. But, the data from the sensors is confined to a small group of people, generally within a utility. In case of a contingency, there is no infrastructure that can be used to share vital information, which can be helpful in preventing outages, saving millions of dollars. Sensor Web Enablement can be thought of as a legacy system friendly infrastructure that can be useful in seamless inter/intra-utility information exchange.

Sensor web provides the necessary information technology infrastructure on which varied decision support tools could be built, as shown in Figure 6. The loosely coupled services in Service Oriented Architecture (SOA) provide the necessary flexibility for cross domain information integration and querying. Web services decouple objects from the platforms that hold them hostage, i.e., web services facilitate interactions among platform-independent objects, which are able to access data from anywhere over the Web. They rely on loose, rather than tight, couplings among web components. Systems that rely on propriety objects are called tightly coupled because they rely on a well-defined but fragile interface. If any part of the communication between applications and service objects is disrupted or if the call is not exactly right, unpredictable results may occur. Loosely coupled systems allow for flexible and dynamic interchange in open, distributed web environments [26].

Sensor Web provides a standard method of discovering and accessing sensor data so that anyone with an internet connection and proper authentication would be able to access them. According to Kevin A. Delin of NASA's Jet Propulsion Laboratory [27], the "Sensor Web concept enables spatio-temporal understanding of an environment through coordinated efforts between multiple numbers and types of sensing platforms, both orbital and terrestrial, both fixed and mobile. Each of these platforms communicates with its local neighborhood of sensors and thus distributes information to the instrument. The Sensor Web is to sensors what the internet is to computers, with different platforms and operating systems communicating by way of a set of robust protocols."

Sensor Web Enablement (SWE) is an initiative taken by the Open Geospatial Consortium (OGC), who has developed a unique and revolutionary framework of open standard for communication with the web-connected sensors of all kinds. The main highlight of the SWE is setting up communications using internet and web protocols, opening up the possibility of web accesses. The use of eXtensible Markup Language (XML) [28] schemas (formal specifications for structured text), providing machine-readable metadata, makes a great step towards automatic sensor monitoring and control. [20]

High Level Architecture

The SWE initiative is more focused towards the development of standards to enable discovery, exchange, and processing of sensor observation. The main functionalities of the power system Sensor Web Enablement, powered by OGC SWE include [29]:

- Discovery of sensor systems, observations, and observation processes that meet an application or users immediate needs;
- Determination of a sensor's capabilities and quality of measurements;
- Access to sensor parameters that automatically allow software to process and geo-locate observations ;
- Retrieval of real-time or time-series observations and coverage in standard encodings ;
- Working of sensors to acquire observations of interest;

The goal of using SWE is to establish a standard communication protocol that can compensate for the heterogeneity of an internal operational and communicational protocol of power utilities for effective communication in case of contingency. The power systems sensor web facilitates the followings:

- The description of power sensors and observation processes with general models and XML encodings through *SensorML* that allows one to fully describe them and hence facilitates the dynamic retrieval of their capabilities and quality of measurements.
- The use of real or near real time data derived from sensors relating to the power systems sensor networks through the Sensor Observation Service (SOS) parameterized

observation requests (by observation time, feature of interest, property, sensor), as illustrated in Figure 6.

- The dynamic selection and aggregation of multiple sensor systems and simulations in web services based environment that provide capabilities for discovering systems, observations, and observation processes that meet an application's or user's immediate needs.

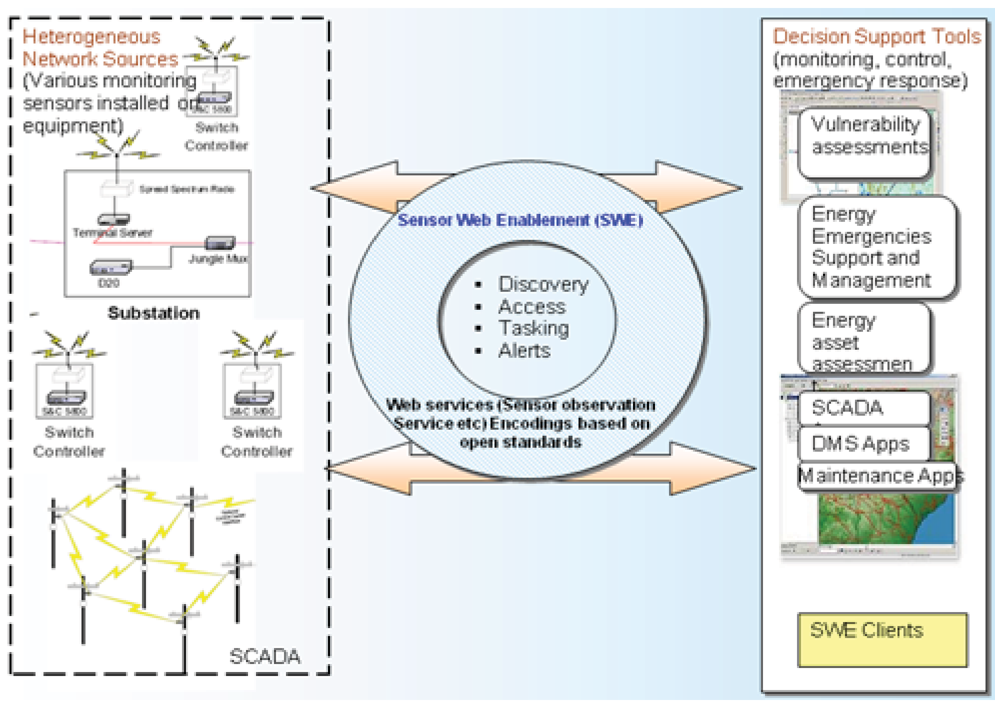

Figure 6 Power Systems Sensor web provides tools for improved decision

The following *OpenGIS*® [www.opengeospatial.org] specifications have been used in the power system Sensor Web development:

i. Sensor Observations Service (SOS) – A standard web service interface for requesting, filtering, and retrieving observations and sensor system information.

This is the intermediary between a client and an observation repository or near real-time sensor channel, thus avoiding the user to access individual sensor.

ii. Sensor Model Language (SensorML) – Standard models and XML Schema for describing power sensors systems and processes. It provides information needed for discovering sensors, locating sensor observations, processing of low-level sensor observations, and listing of workable properties.

iii. Observations & Measurements Schema (O&M) – Standard models and XML Schema for encoding observations and measurements from a sensor.

SWE Standards Framework

The following section describes the SWE specifications used to build the power system sensor web.

Sensor Observation Service

A Sensor Observation Service (SOS) provides an API for managing deployed sensors and retrieving sensor data, specifically "observation" data. SOS acts as an interface for accessing the sensor characteristics and the sensor data to the rest of the world. Instead of communicating with each sensor individually for sensor data, SOS groups sensors into several constellations that can be accessed via SOS [5].There exists variety of sensors from remote sensors to in-situ, fixed to mobile, simple to complex sensors which could be grouped together for a common interface irrespective of the type of the sensor, as illustrated in Figure 7 [5].

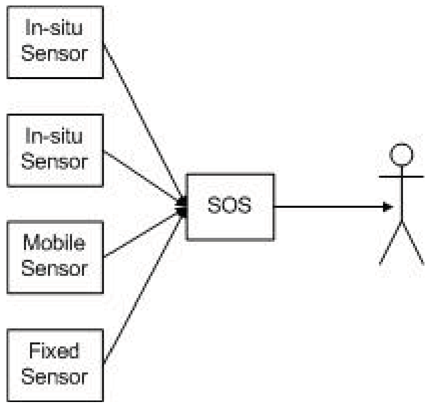

Figure 7 SOS hides the heterogeneity of sensors from Sensor data consumer

SOS as an Organizer

An SOS organizes the collections of related sensor system observations into Observation "Offerings". Each Observation Offering is constrained by a number of parameters including the followings:

- Specific sensor systems that report the observations.
- Time period(s) for which observations may be requested (supports historical data).
- Phenomena that are being sensed.
- Geographical region that contains the sensors.
- Geographical region that is the subject of the sensor observations (may differ from the sensor region for remote sensors).

The OGC Sensor Observation Service specification [*http://www.opengeospatial.org/standards/requests/32*] defines an API for managing deployed sensors and retrieving sensor data, specifically "observation" data. Whether from in-situ sensors (e.g., voltage, current, angles, etc) or dynamic sensors, measurements made from sensor systems contribute most to the geospatial data by volume used in geospatial systems today. SOS implementation specification defines the interfaces and operations that enable the implementation of interoperable sensor observation services and clients. The SOS is the intermediary between a client and an observation repository or near real-time sensor channel. Clients implementing SOS can also obtain information that describes the associated sensors and platforms. The SOS *GetObservation* operation includes an ad-hoc query capability that allows a client to filter observations by time, space, sensor, and phenomena, as shown in Figure 6 of the power systems architecture. Different requests, such as *GetCapabilities, GetObservation,* and *DescribeSensor*, are handled by the SOS.

SOS Operation

SOS-Sensor Data Consumer Interaction

A sensor data consumer is the application or person interested in sensor data. There may be two kinds of approaches of the data consumer to the SOS, a sensor-centric approach and an observation-centric approach [5].

A sensor-centric point of view would be used if the data consumer was already aware of the existence of particular sensors and wants to find observations for those sensors. An observation-centric point of view would be used if the consumer wants to see sensor data from a particular geographic area or particular characteristics that capture particular phenomena but is not aware of any particular sensors a-priori [5].

Figure 8 demonstrates the sequence diagram that shows a sensor data consumer discovering two SOS instances from a CS-W catalog by using the *GetRecords* operation.

The consumer then performs service-level discovery on each service instance requesting the capabilities document and inspecting the observation offerings. The consumer invokes the *DescribeSensor* operation to retrieve detailed sensor metadata in SensorML for sensors advertised in the observation offerings of the two services. Finally, the consumer calls the *GetObservation* operation to actually retrieve the observations from both service instances.

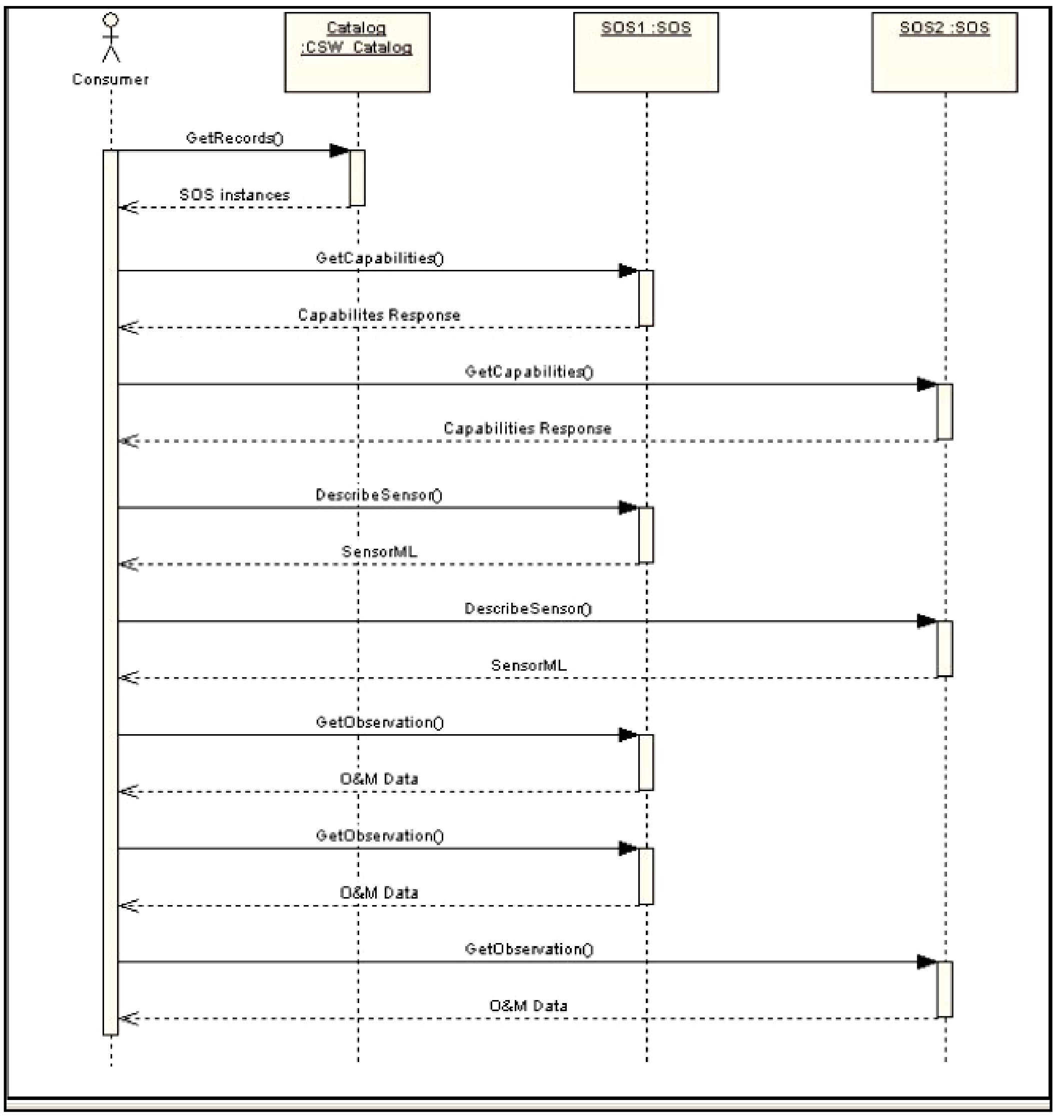

Figure 8 Sensor Data Consumer Sequence Diagram [5]

Get Sensor Metadata

The power system may have spatially distributed sensors. In case of a contingency, it may be required to get sensor data of certain sensors to make some critical decisions. Sensor metadata can be retrieved for any sensor that is advertised in an observation offering using the *DescribeSensor* operation. Each of the sensor characteristics has to be described by the sensor deployer (generally utilities) in the form of SensorML. The SOS will return a corresponding SensorML document with detailed information about the sensor. This document can be used to filter out sensors that do not have robust error detection and correction or are not accurate enough to rely on by the security coordinator for decision making. This is an optional step that can be skipped if the consumer has prior knowledge of the sensor.

Get Sensor Observation

This is an actual operation of receiving sensor data. Sensor observations are obtained using the *GetObservation* operation. This operation supports a query mechanism that supports sub-setting the observations that will be returned from a call to *GetObservation. GetObservation* and allows the client to filter a large dataset to get only specific observations that are of interest. For example, the user can extract the voltage within a certain range only within a certain geographic area. The sub-setting of the sensor data is very useful, especially in power systems, because of an overwhelming number of nodes being monitored at any given time.

Service Discovery

As it is not necessary for each utility to host a SOS, the utilities may need to discover a SOS which can be used to publish information. The service discovery is done by using one or more OGC Catalog Service (CS-W) instances, as described in the data consumer service discovery. The sensor data publishers are likely to be more tightly bound to SOS instances than consumers and are less likely to incorporate a catalog client. Therefore, this step will probably be skipped for most producers and the producers will instead be manually configured with the location of the SOS instance when they are deployed.

Sensor Registration and Publishing of Observations

For data producers, the SWE provides the ability to publish observations to an SOS if that SOS already knows about the sensor that generates the observations. The producer can look at the capabilities document of the SOS to determine whether the sensor is already known to the SOS. If not, then the producer must register the sensor using the *RegisterSensor* operation. Once the sensor is registered with a SOS instance, the data producer can begin publishing observations of the sensor. The SOS is responsible for packaging the observations into offerings and providing them to sensor data consumers. The sensor data has to be packed, such that data packed together are relevant to each other, so that precious time would not be lost at the time of a contingency to find relevant data. In this research, the source of the data comes from a standardized CIM models for the power sensors.

SOS as Sensor Data Repository

The Sensor Observation Service (SOS) maintains a spatial database, which can perform queries based on geographic latitude and longitude, as shown in Figure 9. The database acts as a repository of the data from different sensors [1]. The sensor data producer is responsible for inserting the observation into the SOS database using the *InsertObservation* service of SOS.

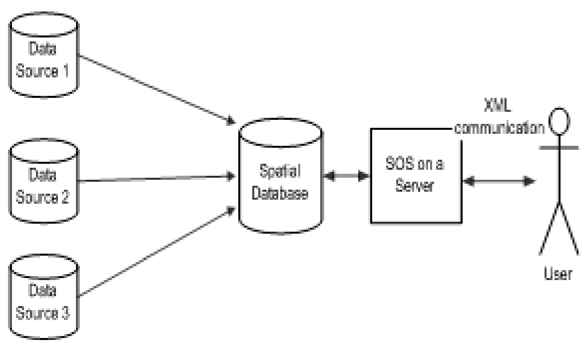

Figure 9 SOS as data repository [1]

The sensor data consumer queries from the SOS database by using *GetObservation* service of the SOS whenever an access to sensor data is necessary. The SOS handles the *GetObservation* query, then accesses the database, forms a Response XML, and sends back as a response to the query, as shown in Figure 9. The data schema with sample data of the SOS database for a power system is shown in Table 1.

Table 1 Sensor Observation Service Database Schema for Power Systems

OBSERVATION		
Columns	**Data to be inserted**	**Sample Data**
Time_stamp	Date/time data collected	12:29:04 PM
feature_of_interest_id	id of power station for which this obervation corresponds	FOI_2001
procedure_id	sensor Model Id	urn:ogc:def:procedure:ifgi-sensor-1
observation_id	*refer observation_value Table*	N/A

OBSERVATION_VALUE		
Columns	**Data to be inserted**	**Sample Data**
obervation_id	Automatically generated	N/A
Phenomenon_id	Phenomenon measured	*refer phenomenon Table*
Value	reading of measured Qty.	50

FEATURE_OF_INTEREST		
Columns	**Data to be inserted**	**Sample Data**
feature_of_interest_id	id of power station	MISS_01
feature_of_interest_name	Name of power station	Station at Starkville
feature_of_interest_description	Address of station	100 presidents circle, MS 39762
geom.	Longitude and latitude of location	200,40

PROC_FOI		
Columns	**Data to be inserted**	**Sample Data**
procedure_id	sensor Model Id	urn:ogc:def:procedure:ifgi-sensor-1
feature_of_interest_id	*refer feature_of_interest Table*	N/A

PHENOMENON		
Columns	**Data to be inserted**	**Sample Data**
phenomenon_id	Id of to be measured electrical Qty.	Current/Voltage
phenomenon_description	Text describing phenomenon	
Unit	Unit of phenomenon	Ampere
Valuetype	type of data	integerType

Table 1 (continued)

PHEN_OFF		
Columns	**Data to be inserted**	**Sample Data**
phenomenon_id	*refer phenomenon Table*	N/A
offering_id	*refer offering table*	N/A

OFFERING		
Columns	**Data to be inserted**	**Sample Data**
offering_id		not specific
offering_name	Name of offering	

PROC_PHEN		
Columns	**Data to be inserted**	**Sample Data**
procedure_id	sensor Model Id	urn:ogc:def:procedure:ifgi-sensor-1
phenomenon_id	*refer phenomenon Table*	

PROCEDURE		
Columns	**Data to be inserted**	**Sample Data**
procedure_id	Phenomenon in data model	urn:ogc:def:procedure:ifgi-sensor-1
procedure_name	Text	Text
procedure_description	Text	Text

Below is the detailed description of each table of the database [30].

1. Observation value table – The observation value table stores the values of an observation event, which is stored in the observation table and its corresponding phenomenon. The actual values of the observation from the sensors, such as voltage and current angles, are stored in this table.

2. Observation table - The observation table aggregates the data of an observation event, such as time, procedure (sensor or group of sensors), the feature of interest and the observation value, which are stored in a separate table. Note that the columns

observation_id, *feature_of_interest_id*, and *procedure_id* are foreign keys. It has to be ensured that the values to be inserted in these columns are contained in the tables they reference on. The time of the retrieval of data, geographical location id of the reading, and sensor id are stored in this table.

3. Feature_of_interest table – The feature_of_interest table stores data about the feature of interest. The *geom* column holds the geometry of the feature_of_interest and is of the PostGIS type geometry. In this table, each of the geographic location is given a specific id with the actual geographic location of the power sensor, thus uniquely identifying each location which may be a substation or a distribution point.

4. Procedure table – The procedure table stores data about the procedure. Only the *procedure_id*, which should be the URN of the procedure as specified by the OGC, must be contained. Each of the power sensors is assigned an id, which uniquely identifies a sensor; the sensor id is called *procedure_id* in this table.

5. Proc_foi table – The proc_foi table realizes the many-to-many relationship between procedures and features of interest. If the new procedures and/or new features of interest are inserted, the relationships have to be taken care of. In this table, each of the power sensors is assigned to a particular geographic location, such as distribution point or substation. If a power sensor is relocated, then this table has to be updated.

6. Phenomenon table – The phenomenon table represents phenomena. In the context of the new SOS specification, phenomena are also called observedProperties. Only the *phenomenon_id* and *value_type* are required. The *phenomenon_id* should contain the URN of the phenomenon as specified by the OGC. The possible values of the *value_type* column are:

•integerType, doubleType, floatType for numerical values

•textType for textual (categorical) values

This table stores a unique id for each of the electrical parameters to be handled by the sensor web, e.g., electrical parameters, such as current, voltage, current phase, and voltage phase, is assigned a unique *phenomenon_id* stored in this table.

7. Proc_phen table – The proc_phen table realizes the many-to-many relationship between procedures and phenomena. If new procedures and/or new phenomena are inserted, the relationships have to be inserted as well in this table. Each sensor has its own capability to measure some specific type of parameters. For example, a current transformer measures current, a potential transformer measures voltage, and PMU can measure multiple parameters. This table establishes a relation between the power sensor and the parameters, the sensor can measure.

8. Offering table / phen_off table – The offering table stores each offering of the SOS. This table is only used when the SOS is initialized to read in the offerings of this SOS (e.g. voltage) and the phenomena which are related to each offering. Each of the phenomena can be bundled together as offering. The phen_off table is created to represent the many-to-many relationship between offerings and phenomena. For Example, voltage and current can be combined together as one offering; "DC Power" while the voltage, current, and phase angle can be combined together as one offering; "AC Power". If new offerings are inserted, the SOS has to be restarted to enable the changes.

The general SOS database schema with all the relationships between tables is illustrated in Figure 10.

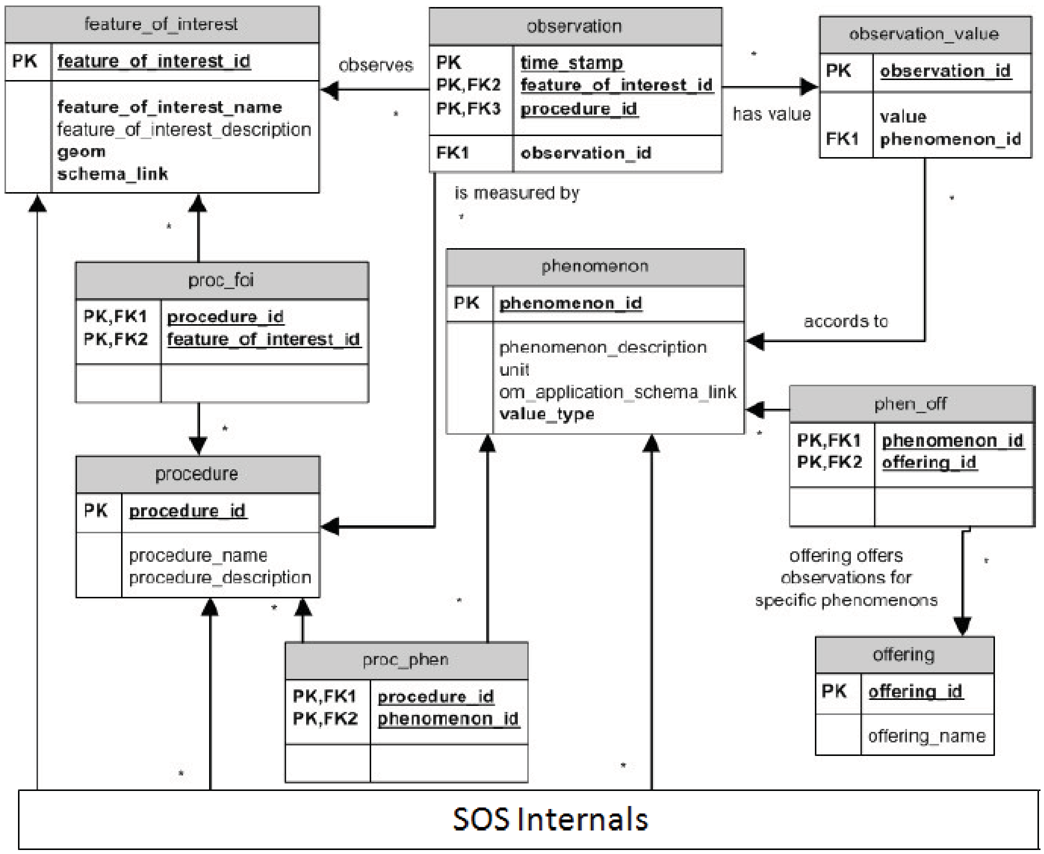

Figure 10 Sensor Observation Service database schema [29]

SensorML

The Sensor Modeling Language (SensorML) is the standard markup language developed by the Open Geospatial consortium (OGC), which provides a common framework for describing characteristics of the sensors. Within SensorML, sensors and transducer components (detectors, transmitters, actuators, and filters) are all modeled as processes that can be connected and participate equally within a process chain. It utilizes the same process model frame as any other process [4]. There are a wide range of sensors being used in the real world, from in-situ anemometers to remote spectral radiometers. All the sensors can be modeled in SensorML with equal expressiveness.

In the context of power applications, sensors, such as Phasor Measurement Unit (PMU), Potential Transformer, Current Transformer, etc. are modeled in SensorML in order to get the characteristics of the sensors, such as accuracy, to find reliability of sensor data, and to make crucial decision in making decision to avoid contingency.

SensorML is a foundational part of OGC's Sensor Web Enablement (SWE). It provides a functional model of the sensor system, rather than a detailed description of its hardware, as shown in Figure 11. The root for all SensorML documents is sensor, or an extension of sensor, such as SensorGroup. The Sensor has a unique id (of type xs:id) as a required attribute. Two optional attributes, documentDate (xs:dateTime), and documentVersion (xs:string), provide the ability to quickly check version information. The sensor description is divided into nine main informational components, each of which has sub-parts. Several of the components have "plug-n-play" capabilities, such that they can accept models that are appropriate for a given class of sensors.

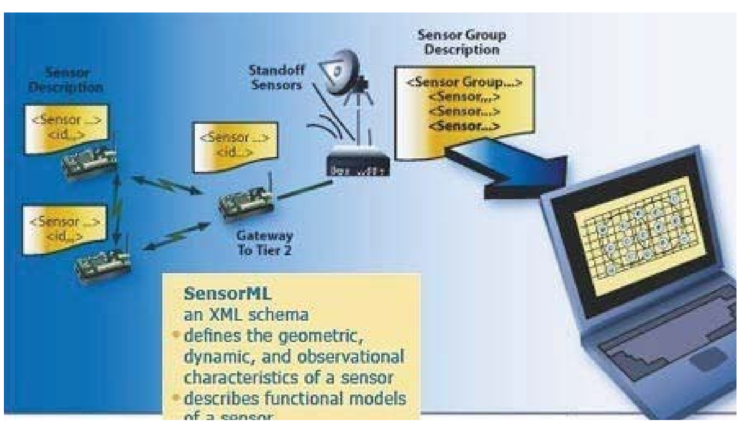

Figure 11 Description of sensors related to power systems in SensorML [42]

SensorML provides a standard schema for metadata that describes sensors and sensor system capabilities. SensorML treats sensor systems and a system's components (e.g. sensors, actuators, platforms, etc.) as processes. Thus, each component can be included as parts of one or more process chains that can either describe the lineage of observations or provide a process for geo-locating and processing the observations higher level information.

The usefulness of the SensorML document can be summarized in the following points:

•Discovery of Sensor data: In addition to inputs, outputs, parameters and system location, SensorML provides a wide collection of metadata that can be used to discover the sensor system and the data. Any sensor can be discovered and queried without even a prior knowledge of the existence of the sensor. The metadata includes identifiers, classifiers, constraints (time, legal, and security), capabilities, characteristics, contacts, and references.

•Lineage of Observations: SensorML can provide the lineage of observations. The SensorML contains information about the input, confidence level, accuracy of the sensor, and even the accuracy curve of the sensor, which gives it a strong backtracking of the process from acquisition to analysis of the sensor data if needed.

•Formal Sensor Description: SensorML is exchanged in XML document, which makes it self-describing and can be processed by an automated system [31].

•Reconfigurable: The addition or removal of a sensor from a system does not require any reconfiguration. All the sensors are plug-N-play provided a valid SensorML document is provided [31].

•Support the Geo-Spatial Data: The measured data can be tied with spatial locations.

•Performance characteristics: The sensor characteristics, such as accuracy and threshold, can be specified in the SensorML to provide analysts with an extra arm for the verification of the sensor data.

• Archiving of Sensor Parameters: SensorML provides a mechanism for archiving fundamental parameters and assumptions regarding sensors and processes so that observations from these systems can still be reprocessed and improved long after the origin mission have ended. This is proving to be critical for long-range applications, such as global change monitoring and modeling [32].

Figure 12 illustrates the snippet of SensorML describing a Phasor Measurement Unit (PMU) sensor. The highlighted terms represent the properties of sensor represented in an XML document, namely SensorID, Sensor Type etc. This document describing the sensor is hosted by SOS, which is sent to the user when requested. See Appendix A for more detailed SensorML.

```xml
<System id=" SensorID ">
          <identification>
                    <IdentifierList>
                              <identifier name="longName">
                                        <Term> Sensor Type </Term>
                              </identifier>
                              <identifier name="shortName">
                                        <Term>PMU</Term>
                              </identifier>
                              <identifier name="modelNumber">
                                        <Term qualifier="urn:ogc:identifier:modelNumber"> Model Number </Term>
                              </identifier>
                              <identifier name="manufacturer">
                                        <Term qualifier="urn:ogc:identifier:manufacturer"> Manufacturer Name </Term>
                              </identifier>
                    </IdentifierList>
                    <inputs>
                              <InputList>
                                        <input name=" Parameter Measured ">
                                         <swe:Quantity definition="urn:ogc:def:phenomenon:OGC:Voltage"/>
                                        </input>
                              </InputList>
                    </inputs>
</System>
```

Figure 12 Snippet of the SensorML of a Power Sensor

Observation and Measurement

Observation and Measurement (OM) requests furnish sensor data to the user. The user sends an OM XML request to the SOS server, which returns an XML response to the user with the data requested. The data has to be resided in the spatial database of the SOS.

This is the standard form of communication between the service provider and the service consumer which is important for the property of interoperability of SWE. Each of the service providers uses the same standard for fulfilling the request of the user, and then the user just needs to be updated with the universal standard to communicate with all the service providers.

The key properties associated with an observation are [33]

- Feature of interest
- Observed property

- The procedure
- The result

The Feature of Interest (FOI) is a feature of any type representing the observation target and being the real world object regarding which observation is made. FOI associates the observation to some kind of real world object, such as geographic area, pixel etc. [33]

The observed property identifies or describes the phenomenon for which the observation result provides an estimate of its value. It must be associated with the type of the feature of interest. For example, if a voltage sensor measures the voltage across a terminal, then the observed property and FOI cannot deliver any information to the end user. If the FOI and the voltage measured are combined and presented to the user as "Voltage across terminal A", then it makes sense to the user. So, Voltage is the observed property while terminal A is the Field of interest.

The procedure is the description of a process used to generate the result. It must be suitable for the observed property.

The result contains the value generated by the procedure. The type of the observation result must be consistent with the observed property, and scale or score for the value must be consistent with the quantity or category type.

CHAPTER IV

IMPLEMENTATION DETAILS

In addition to the Sensor Web Enablement standards, several supporting software tools have been used for the prototype development. In this chapter, the implementation details of the prototype, which are necessary to replicate the project, are discussed.

Servlet Container (Apache Tomcat)

Apache Tomcat is an implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the Java Community Process [34].

The power systems Sensor Observation Service servlet resides in the Apache Tomcat container and it is responsible for serving the request/response from the client. This acts as a middleware between the sensor data and the client.

The power system sensor web uses Tomcat version 5.5, which can be downloaded from (http://tomcat.apache.org/download-55.cgi). Java 5 or above must be installed in the system running the Tomcat. The only configuration to be set for running tomcat in case of windows is setting an environment variable "JAVA_HOME", to determine the base path of a J2SE 5 Java Running Environment (JRE). See http://tomcat.apache.org/tomcat-5.5-doc/setup.html for details of the platform specific configuration.

The Power Sensor Observation Service has to be deployed in Apache Tomcat as a web application archive (WAR) file. The SOS contains all the building information in /conf folder. See [29] for details of configuring SOS and Tomcat.

Querying the CIM Resource Description Framework (RDF)

Simple Protocol and RDF Query Language (SPARQL), pronounced as "sparkle", is the key standard for opening up data on the Semantic Web developed by the World Wide Web Consortium (W3C). SPAQRL can query on the RDF document similar to SQL querying on a database. SPARQL contains a standard query language, a data access protocol, and a data model [35].

Conventional query languages, such as SQL, are confined to a single product, format, and type of information, or local data store, which require formulating some kind of logic if multiple data sources come into a scene. This conventional data accessing mechanism does not match with the evolving semantic web, which aims to enable sharing, merging, and reusing data globally [36].

SPARQL empowers users to query variant data sources with different data formats with same queries.

The power sensors monitor the power system continuously, thus providing continuous readings. Sensor data is continuously published in the form of CIM in a preset interval. The continuously updated sensor data described in the CIM representation should be queried each time the data changes dynamically and also the spatial database of the SOS should be updated correspondingly. To address this problem, the SPARQL query language, which is a RDF querying mechanism standardized by W3C, has been adopted. This allows gleaning the instance data from the CIM easily and updating the SOS database when required. To achieve the above objectives, it is required to configure a SPAQL querying server whose details are described below.

JOSEKI Server

JOSEKI is an HTTP engine that supports the SPARQL Protocol and the SPARQL RDF Query language developed by Hewlett Packard [38]. JOSEKI acts as a service to the user querying on the RDFs. The user sends a SPARQL request to query on the RDFs to JOSEKI, which, in turn, actually makes the query on the RDF(s) and returns to the user the result in the form of an XML document, as illustrated in Figure 13. This makes the query of the RDF document no more than a web service access, which significantly decreases the time of developers.

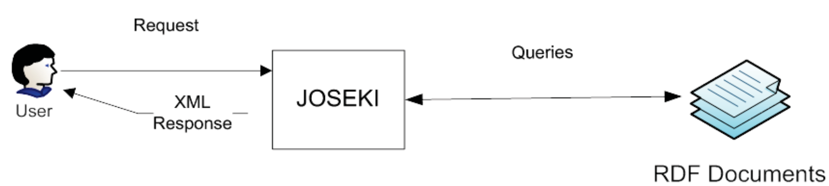

Figure 13 JOSEKI-User Interactions

Executing SPARQL Queries

Most forms of the SPARQL query contain a set of triple patterns called a basic graph pattern. Triple patterns are similar to RDF triples except that each of the subject, predicate, and object, may be a variable. A basic graph pattern matches a sub graph of the

RDF data when the RDF terms from that sub graph may be substituted for the variables and the result is a RDF graph equivalent to the sub graph.

The following query shows a trivial form of SPARQL query. It illustrates the query for determining the time stamp of a measurement from a CIM RDF. The SELECT clause consists of the variable *'?timestamp'* to be appeared in the query results and the WHERE clause provides the basic graph pattern to be matched against the data graph. Appendix B contains a sample SPARQL query.

Data:

<cim:MeasurementValue:timestamp> 6/06/2008 01:03:02 PM

</cim:MeasurementValue:timestamp>

Query:

SELECT ?timeStamp FROM http://www.urladdresss.com/voltage.xml

WHERE

{

?s cimMeasurement.timeStamp ?timeStamp.

}

Result:

timestamp = "6/06/2008 01:03:02 PM"

CHAPTER V

RESULTS

Using the standards described in the previous chapters, the AJAX client prototype was developed in the Google Web Toolkit (GWT) [39]. The details of the prototype application are described below. The application has been named as *PowerPicket*, where picket means a person employed to keep watching some unanticipated events [40]. Thus, *PowerPicket* is an appropriate name for the application to monitor the power systems.

Sensor Web Client

PowerPicket is developed using the Google Web Toolkit (GWT). The client is the prototype of the application that security coordinators will interact with to monitor their regions.

The flow of data within the applications is illustrated in Figure 14. The utilities within a region share the data in the form of CIM/XML. The CIM/XML is queried using SPARQL to extract relevant data, which is then inserted into a spatial database. The spatial database acts as a central repository of the sensor data of all the utilities irrespective of the ownership of the data. Also, several SOSs can be developed by independent agencies and registered in a central catalog to discover and access a relevant SOS for a particular application.

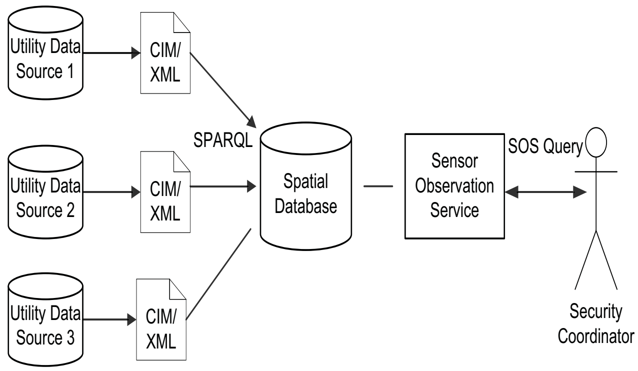

Figure 14 Data Flow in *PowerPicket*

The data can be queried with SOS standard queries using the client application to monitor an entire region. The client application is powered with Google Maps™, which can geographically locate each sensor in the entire region empowering the security coordinator to act more precisely against a contingency.

The user-interface (UI) of *PowerPicket* is shown in Figure 15. The user has to follow the following steps to get the data from the SOS database in a typical case.

1. The user selects the offerings in which he/she is interested in. The UI is marked by '1' in Figure 15.

2. He/she selects the sensor ID he/she is interested in from the list of available sensors. The UI is marked as '2' in Figure 15.

3. The user can perform the service level discovery on each service available by requesting capabilities document by pressing a button marked as '3' in Figure 15.

4. The user can then query for power sensor data by pressing a button marked as '5' in Figure 15. The application sends a *DescribeSensor* request to get the SensorML describing the characters of a particular sensor whose data has been asked by

the user. The SensorML document and the capabilities documents are shown in the blowup window in the map with the marker representing the sensor.

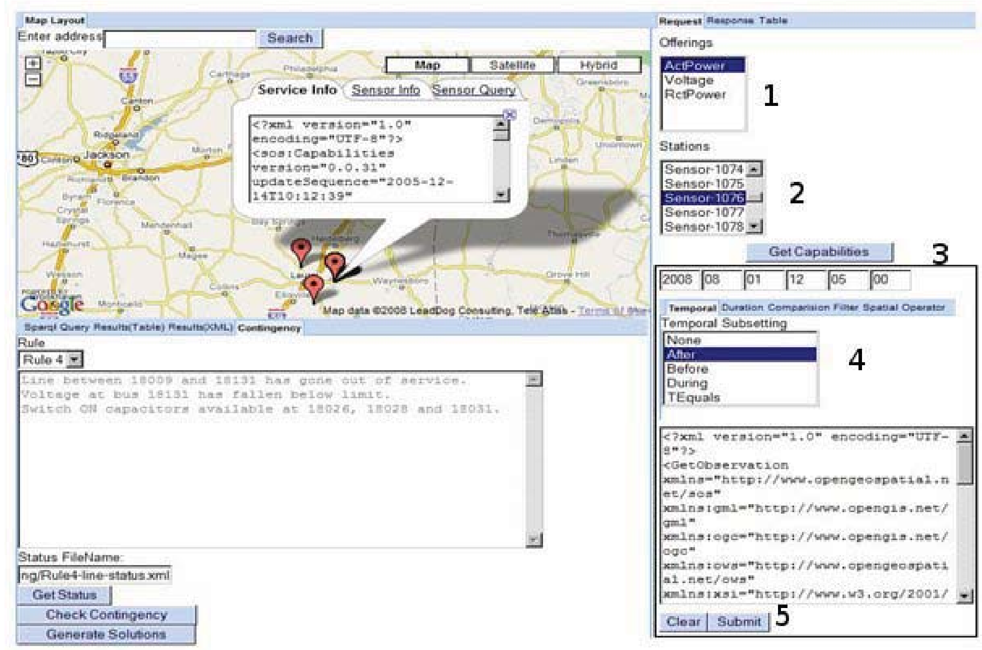

Figure 15 User Interface of the *PowerPicket*

The Sensor Observation Service (SOS) supports a number of spatio-temporal querying of sensor data [5]. The user can query for the data as per his/her requirement of analysis. In Figure 15, the tabs in the portion labeled ‘4’ are useful in requesting different SOS queries. These are described in detail in the following sections.

Spatial Subsetting

The queries that are supported by this operation include selecting a bounding box and retrieving data from all the sensors within the selected region. In addition to getting the sensor data within a region, parameters, such as the overlapping of two regions, containing a certain point, and intersection of two regions, are also supported for filtering sensor data. These kinds of spatial queries are enabled by the integration of a spatial database engine (e.g. PostGIS) to RDBMS. Figure 16 shows the snippet of the XML request that is actually sent to the SOS. Figure 17 shows the portion of Figure 15 that is used for the spatial subsetting.

```
<offering>Potential</offering>
<observedProperty>urn:ogc:def:phenomenon:voltage</observedProperty>
<featureOfInterest>
<ogc:BBOX>
<gml:Envelope srsName= "EPSG:4326">
<gml: lowerCorner>29.6880527498568 92.5048828125</gml: lowerCorner>
<gml: upperCorner>37.02009820136811 102.1728515625</gml:upperCorner>
</gml:Envelope>
</ogc:BBOX>
</featureOfInterest>
<resultFormat>text/xml;subtype="OM"</resultFormat>
</GetObservation>
```

Figure 16 Snippet of a spatial query of the sensor web for retrieving data

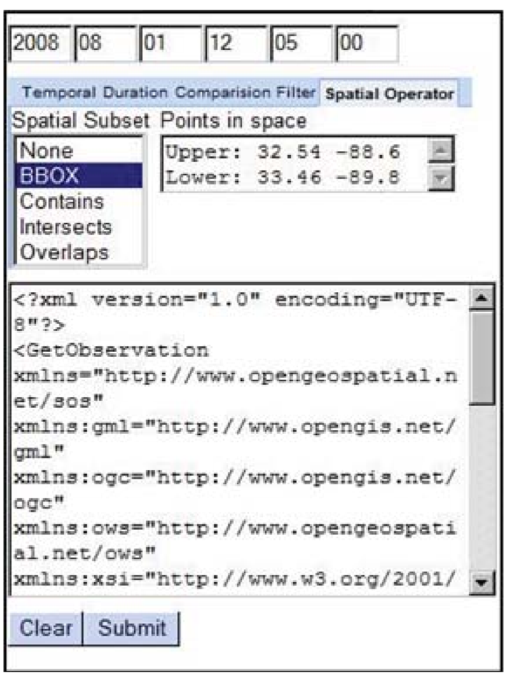

Figure 17 User Interface used for spatial subsetting

Temporal Subsetting

Temporal subsetting represents the subsetting of the existing sensor data with respect to a time instant, such as after a time instant, before a time instant, during a time instant, past hour, day, minutes, etc. This type of subsetting is very useful in requesting the historian data of the power systems. The snippet shown in Figure 18 is an example of the actual temporal query sent to SOS. Figure 19 shows the two windows that are used to create temporal queries. The left window in Figure 19 is used to create queries for after, before, during, and/or at a certain time instant while the right window is used to create temporal queries by specifying duration (data for past 5 days).

```
<offering>Potential</offering>
<! -- eventTime is time of past 5 days-->
<eventTime>
<ogc:During>
<gml:TimePeriod>
<gml:beginPosition
indeterminatePosition="unknown"></gml:beginPosition>
<gml:endPosition>2006-10- 10T10:00:00</gml:endPosition>
<gml:duration>P5D</gml:duration>
</gml:TimePeriod>
</ogc:During>
</eventTime>
<observedProperty>urn:ogc:def:phenomenon:voltage</observedProperty>
<resultFormat>text/xml;subtype="OM"</resultFormat>
</GetObservation>
```

Figure 18 Snippet of a temporal query for retrieving data from within past 5 days

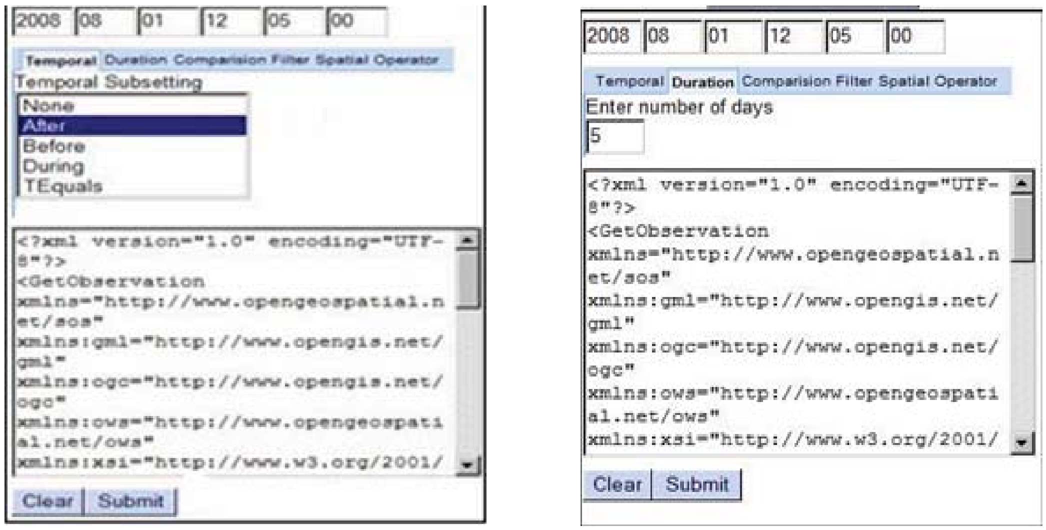

Figure 19 User Interface used for Temporal subsetting

Filtering

SOS supports filtering of the data with respect to their values. The data, which are within a certain range of numeric values by using operators, such as Between, EqualTo, NotEqualTo, LessThan, and GreaterThanEqualTo, can be requested from SOS, as shown in Figure 20. The actual snippet of the XML request sent to SOS is shown in Figure 21.

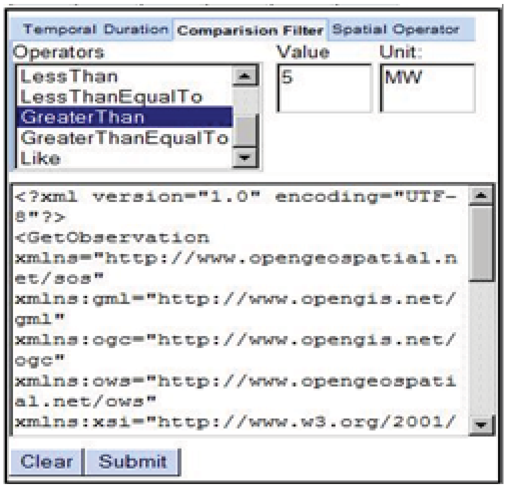

Figure 20 User Interface for creating comparison filtering query

```
<offering>potential</offering>
<observedProperty>urn:ogc:def:phenomenon:voltage</observedProperty>
<!--result filter filters the observation values which equal to 1000V -->
<Result>
<ogc:PropertyIsLessThan>
<ogc:Literal>
<ogc:Measure uom="volts">1000</ogc:Measure>
</ogc:Literal>
</ogc:PropertyIsLessThan>
</Result>
<resultFormat>text/xml;subtype="OM"</resultFormat>
</GetObservation>
```

Figure 21 Snippet of a comparison filter query of power sensor web

The response to all these queries is sent by the SOS to the client in the form of XML document. A snippet of the XML document is shown Figure 22.

```xml
<om:ObservationCollection gml:id="ACTIVE_POWER090108152904" xsi:schemaLocation="http://www.opengis.net/om
http://mars.uni-muenster.de/swerep/trunk/om/1.0.30/om.xsd">
<om:Measurement gml:id="ID12306" xsi:type="om:MeasurementType">
<om:time>
        <gml:TimeInstant xsi:type="gml:TimeInstantType">
                <gml:timePosition>2008-08-01T12:00:00</gml:timePosition>
        </gml:TimeInstant>
</om:time>
<om:location/>
<om:procedure xlink:href="urn:ogc:def:procedure:-Sensor-1000"/>
<om:observedProperty  xlink:href="urn:ogc:def:phenomenon:OGC:ActPower"/>
<om:featureOfInterest xlink:href="FOI_261">
        <om:Station xsi:type="om:StationType">
<om:position>
<gml:Point srsName="EPSG:4326">
        <gml:coordinates>31.258984 98.008</gml:coordinates>
</gml:Point>
</om:position>
```


Figure 22 Snippet of the response to the Queries

Alert System

The *PowerPicket* is also capable of alerting the user about contingencies based on the offline analysis. The details of the rules and the possible remedial action are discussed below.

Rules for addressing contingencies detection and proposing remedial action

In addition to the querying on the SOS for the sensor data for decision making, five rules, that provide details about contingencies, have been embedded in the system. The rules also contain some basic measures to assist the user with a possible correction measure. A sample rule is shown in Figure 23. The flowchart to check for contingencies with embedded rules is shown in Figure 24.

If voltage in Sensor 1 drops below “threshold” then open switch 5.

Figure 23 Sample Rule with correction Measure

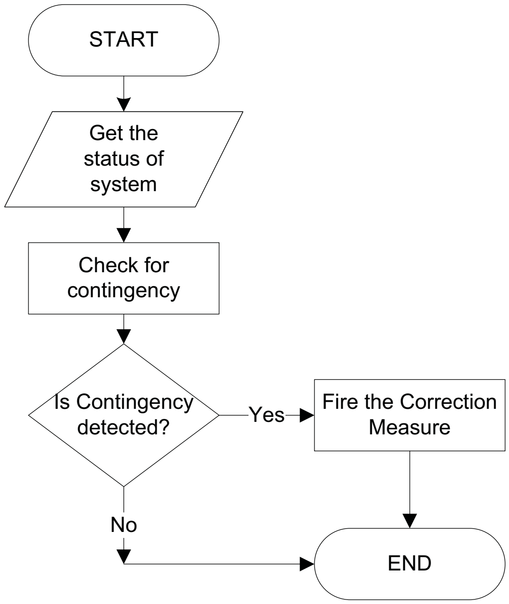

Figure 24 Flowchart to check for the contingency

For example, if a voltage, detected by Sensor 1, Sensor 3, and Sensor 5, goes below 1KV and the power detected by Sensor 6 goes higher than 1KVAR, then closing Breaker 1, Breaker 4, and Breaker 5 may help in avoiding an outage.

The system checks for the current status by querying on the CIM/RDF with a SPARQL query. If all the voltages detected by Sensor 1, Sensor 3, and Sensor 5 have voltages above 1KV and the power detected by Sensor 6 is lower than 1KVAR, then it signifies that everything is normal in the system. The user is notified by overlaying a marker in the geographic location of the sensor in Google Maps™, as shown in Figure 25. The marker  is used to signify normal readings of the sensors. If any anomaly is detected in the sensor measurement, as per the rule, then, the marker  is used to signify a contingency, as shown in Figure 26.

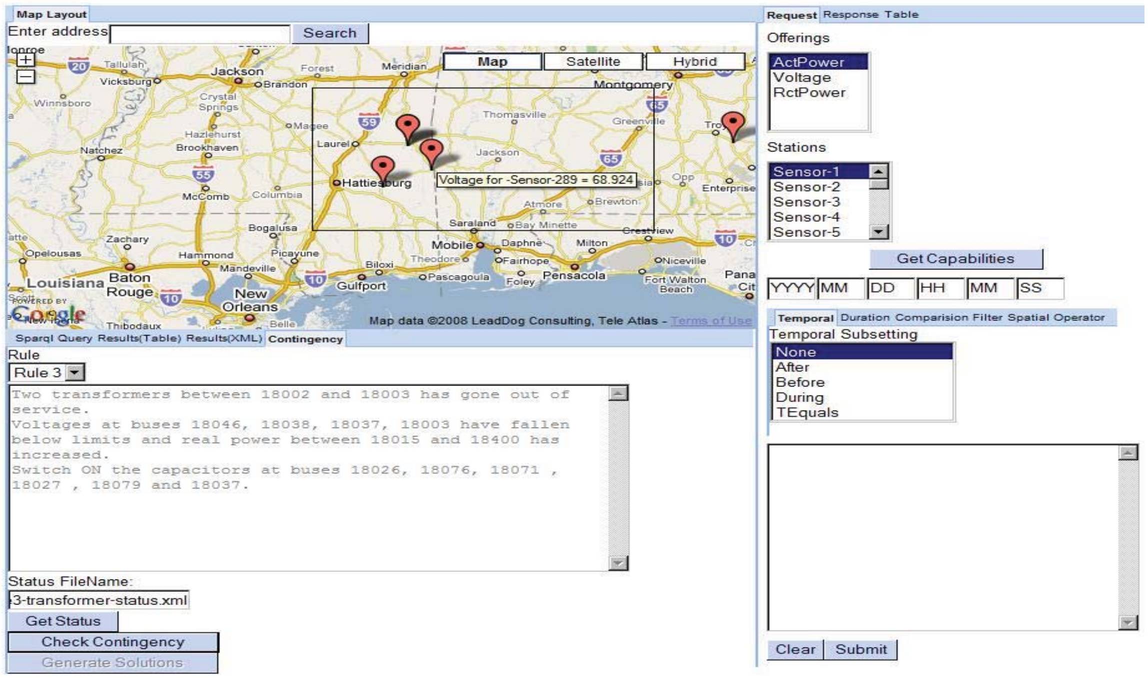

Figure 25 User Interface for no contingency detected

(The icon in the Map shows the geographic location of the Sensor, with the reading shown when the mouse pointer hovers over the icon. The Status of the Grid has to be provided in the form of CIM whose location has to be given as status file name)

The rule also contains some correction measures, as shown in the consequent part of the example rule. The closing of Breaker 1, Breaker 4, and Breaker 5 are the correction measures that have been devised by electrical engineers. The correction measures are displayed in the Google Maps with marker  overlaid in geographic position of the respective breakers.

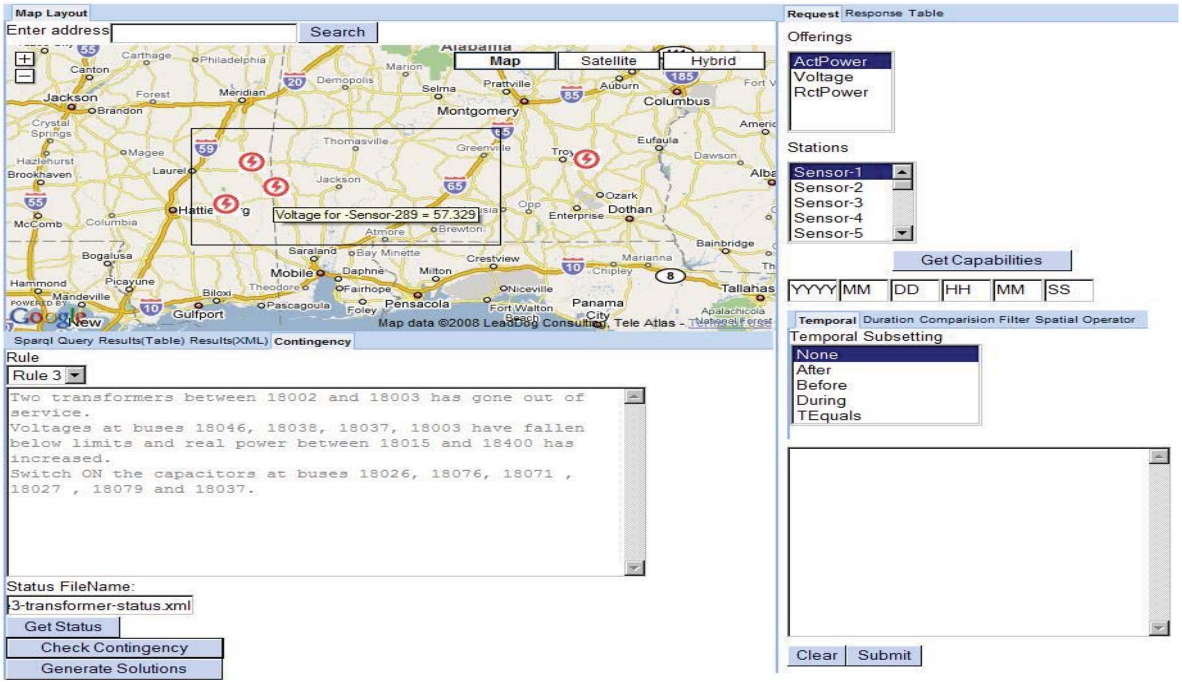

Figure 26 User Interface for contingency detected

(The icon in the Map changes to indicate the violation in the respective sensor, with additional information showing when user hovers mouse pointer over the icon)

CHAPTER VI

CONCLUSION AND FUTURE WORK

The wide range of applications used within the power industry is not compatible with each other, which is hampering the inter-utility communication. A series of coordinated actions over a wide area, which is essential to prevent an outage, is not possible because of time consuming and often tedious inter-utility communications. In recent history, a number of blackouts, which can be prevented with timely and proper information exchange, can be found. The blackouts were unavoidable because of lack of infrastructure for a coordinated action by security coordinators. Currently, the security coordinators are able to act based only on local intelligence, which may not correctly give the global status of the power system. These actions are often uncoordinated and sometimes even counter to each other.

In this study, standards, such as Common Information Model (CIM), Sensor Observation Services (SOS), Sensor Modeling Language (SensorML), and Observation and Measurements (OM), have been used as inter/intra utility information exchange format. A central repository of sensor data serves as common intelligence for all thelocal actions taken to prevent an outage. This makes the local actions well coordinated and ensures that the actions taken are against the spreading of contingencies.

A prototype application based on the Sensor Web Enablement (SWE) framework has been developed. The prototype is especially designed for operators and security coordinators to access critical information about the health of the power system from the central intelligence database. The SOS encapsulates all the complexities of the database and sensors by just exposing the sensor data to the user. The access of the sensor data, with SOS, is as easy as browsing a website. The user will also have leverage to access the properties of the sensor before making any critical decision based on sensor data. For example, before making a critical decision, the user may want to know the accuracy of the sensor. The SOS provides the sensor properties via an OGC standard SensorML.

Future Works

Currently, the prototype application does not have extensive analysis of the information obtained from the utilities. In the future, the prototype application can be enhanced to include more accurate and specific alerts, with suggestions for directly attacking the problem. The application can be equipped with tools, such as state estimation, voltage stability analysis and line temperature monitoring Remedial Action Scheme (RAS) for more reliable alert and warning system.

Currently, RDF has been adopted as the format for Common Information Model. The RDF can be replaced by OWL, which is more expressive than RDF for better knowledge representation. The adoption of OWL as the format of information exchange opens a door for the concept of automatic operator. The contingency detection rules and recovery rules, which have been coded as traditional "if-then" rules in *PowerPicket*, can be expressed as Semantic Web Rule Language (SWRL) [41]. This will give a different paradigm to the reasoning in knowledge representation, bolstering the reasoning system of power system applications.

The ultimate extension of this research would be to implement semantic web, which can match/merge ontology from many sources to build a common intelligence irrespective of the publisher of the ontology. The common intelligence would be supported by a reasoning system, such as SWRL, which can run on the merged ontology to decide on the actions to be taken in cases which may or may not have been anticipated by the designer of the grid topology.

REFERENCES

[1]. N. Dahal, V.M. Mohan, S.S. Durbha, A. Srivastava, R.L. King, N.H. Younan, N.N Schultz, “Wide Area Monitoring using Common Information Model and Sensor Web,” 2009 Power Systems Conference and Exposition, March 15-18, 2009, Seattle, Washington, USA.

[2]. Ralph Mackiewicz, “The impact of Standardized Models, Programming, Interfaces and Protocols on Substation, “http://www.sisconet.com/downloads/Models%20APIs%20and%20Protocols%20White%20Paper%20and%20Presentation.pdf

[3]. Sensor Web Enablement, http://www.opengeospatial/org/projects/groups/sensorweb

[4]. OGC® SensorML, http://www.opengeospatial.org/standards/sensorml/

[5]. Sensor Observation Service (SOS), http://www.opengeospatial.org/standards/sos

[6]. Distributed Management Task Force, Common Information Model (CIM) Standards, http://www.dmtf.org/standards/cim/

[7]. Marek Zima, Mats Larsson, Petr Korba,Christian Rehtanz, and Göran Andersson, “Design Aspects For Wide-Area Monitoring And Control Systems,” Invited Paper, Proceedings of the IEEE, vol. 93, no. 5, May 2005.

[8]. Miroslav Begovic, Damir Novosel, Daniel Karlsson, Charlie Henville, Gary Michel, “Wide-Area Protection and Emergency Control,” Invited Paper, Proceedings of the IEEE, vol. 93, no. 5, May 2005. [9]. J. Bertsch, M. Zima, A. Suranyi, C. Carnal, C. Rehtanz, M. Larsson, “Experiences with and Perspective of the System for Wide Area Monitoring of Power Systems, Quality and Security of Electric Power Delivery Systems,” 2003 CIGRE/IEEE PES International Symposium.

[10]. TCP/IP Tutorial, http://www.w3schools.com/tcpip/default.asp

[11]. Joseph M. Weiss, “Control Systems Cyber Security— The Current Status of Cyber Security of Critical Infrastructures,” a testimony before the Committee on Commerce, Science, and Transportation U.S. Senate, March 19, 2009.

[12]. A. Creery, E.J. Byres, Industrial Cyber-Security for Power Systems and SCADA Networks.

[13]. Rene Avila-Rosales, Jay Giri, Wide-Area Monitoring and Control for Power System Grid Security.

[14]. Manu Parashar, Jianzhong (simon) Mo, “Electric Power Group, Real Time Dynamics Monitoring System (RTDMS TM): Phasor Applications for the Control Room,” Proceedings of the 42nd Hawaii International Conference on System Sciences 2009.

[15]. G. Zhang, P.Hirsch, S. Lee, “Wide Area Frequency Visualization using Smart Client Technology,” Power Engineering Society General Meeting, 2007. IEEE, 24-28 June 2007

[16]. Donnelly, M. Ingram, M. Carroll, J.R., “Pacific Northwest National Laboratory, Eastern Interconnection Phasor Project,” Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006.

[17]. J. Bertsh, M. Ingram, L. Broski, W. Sattinger, “Experiences and Benefits of Systems of Wide Area Monitoring,” Western Power Delivery Automation Conference (WPDA) Spokane, United States of America, 6th to 7th April 2004,

[18]. T. Babnik, U. Gabrijel, B. Mahkovec, M. Perko, and G. Sitar, “Wide Area Measurement System in Action,” 2007 IEEE Power Tech, Lausanne, Switzerland.

[19]. http://wordnetweb.princeton.edu/perl/webwn

[20]. Man Qian, Guo Jinzhi, Yang Yihan, “An Ontology for Power System Operation Analysis ,” 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, April 2004, Hong Kong.

[21]. Dy-Liacco, T.E., “Enhancing Power System Security Control,” IEEE Computer Applications in Power, Volume 10, Issue 3, Pages: 38-41, Jul 1997.

[22]. Mathias Uslar, “Semantic Web Technologies for Power Grid Management, “ INFORMATIK 2007 Informatik trifft Logistik Band 1 Beiträge der 37. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 24. - 27. September 2007 in Bremen.

[23]. Jay P. Britton and Arnold N. deVos, "CIM based Standards and CIM Evolution," IEEE transaction on Power Systems VOL. 20, NO 2, MAY 2005.

[24]. Stephen Quirolgico, Pedros Assis, Andrea Westerinen, Michael Baskey, Ellen Stokes, "Towards a Formal Common Information Model Ontology," Lecture Notes in Computer Science, Volume 3307/2004, Pages 11-21.

[25]. http://www.w3schools.com/rdf/rdf_owl.asp

[26]. Nicolas Younan et. al, "Semantics-Driven Knowledge Discovery System for Wide Area Monitoring of Electric Power Grid," Final Report, January 2009.

[27]. The Sensor Web's Point of beginning, Mark Reichardt, http://www.geospatial-solutions.com/ geospatialsolutions article/articleDetail .jsp?id=5268 1

[28]. Extensible Markup Language (XML) http://www.w3.org/XML

[29]. Installation Guide for Sensor Observation Service version 2-01-00.

[30]. OGC Sensor Web Enablement: Overview And High Level Architecture,http://portal.opengeospatial.org/files /?artifact_id = 25562

[31]. A. Triantafyllidis, V. Koutkias, I. Chouvarda and N. Maglaveras , "An Open and Reconfigurable Wireless Sensor Network for Pervasive Health Monitoring," Proceedings of Second International Conference on Pervasive Health 2008, Jan. 30 2008-Feb. 1 2008 Page:112 - 115.

[32]. Introduction to SensorML, http://vast.uah.edu/index.php?option=com_content&view=article&id= 14& Itemid=52/

[33]. Observations and Measurements, http://www.opengeospatial.org/standards/om

[34]. Apache Tomcat, http://tomcat.apache.org

[35]. SPARQL Query, http://www.w3.org/TR/rdf-sparql-query

[36]. W3C opens Data on the web with SPARQL, http://www.w3.org/2007/12/sparql-pressrelease

[37]. Google Web Toolkit, http://code.google.com/webtoolkit/

[38]. http://www.joseki.org

[39]. Google Web Toolkit, http://code.google.com/webtoolkit/

[40]. The Free Dictionary, http://www.thefreedictionary.com/picket

[41]. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/2004/ SUBM-SWRL-20040521/

[42]. http://www.mitre.org/news/the_edge/spring_06/los.html

APPENDIX A

SENSORML

APPENDIX A

SENSORML


```
<?xml version="1.0"?>
<SensorML xmlns="http://www.opengis.net/sensorML"
xmlns:swe="http://www.opengis.net/swe"
   xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xsi:schemaLocation="http://www.opengis.net/sensorML

file:///C:/Users/nd54.ECE/Desktop/senML/senML/SensorML/sensorML/1.0.30/base/sens
orML.xsd"
   version="1.0">

   <!-- First current transformer parameters of PMU -->

   <Sensor id="CT_001">
      <description>
         <swe:Discussion>Current transformer SensorML </swe:Discussion>
      </description>

      <parameters>
         <ParameterList>
            <parameter name="PrimaryWinding">
               <swe:Count>10</swe:Count>
            </parameter>

            <parameter name="PrimaryWidingPhase">
               <swe:Category>A</swe:Category>
            </parameter>

            <parameter name="PrimaryWindingTerminal1">
               <swe:Category>#100</swe:Category>
            </parameter>

            <parameter name="PrimaryWindingTerminal2">
```

```
   <swe:Category>#102</swe:Category>
</parameter>

<parameter name="SecondaryWinding">
   <swe:Count>10</swe:Count>
</parameter>

<parameter name="SecondaryWindingPhase">
   <swe:Category>AN</swe:Category>
</parameter>

<parameter name="SecondaryWindingTerminal1">
   <swe:Category>#101</swe:Category>
```

```
        </parameter>

        <parameter name="SecondaryWindingTerminal2">
          <swe:Category>#102</swe:Category>
        </parameter>

        <parameter name="CoolingType">
          <PhysicalProperties>
            <material>
              <swe:Category>Oil</swe:Category>
            </material>
          </PhysicalProperties>
        </parameter>

        <parameter name="TransformerType">
          <swe:Category>Tank</swe:Category>
        </parameter>

        <parameter name="Transformerphases">
          <swe:Category>A</swe:Category>
        </parameter>

        <parameter name="PrimaryWindingTerminal1">
          <swe:Category>#100</swe:Category>
        </parameter>

        <parameter name="PrimaryWindingTerminal2">
          <swe:Category>#102</swe:Category>
        </parameter>

        <parameter name="SecondaryWindingTerminal1">
          <swe:Category>#101</swe:Category>
        </parameter>

        <parameter name="SecondaryWindingTerminal2">
          <swe:Category>#103</swe:Category>
        </parameter>

      </ParameterList>
    </parameters>
  </Sensor>

  <!-- This is the SensorML description for Phasor Measurement Unit -->
  <System id="PMU_001">

    <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
```

```
<!-- Description of the Sensor Used       -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<description>
   <swe:Discussion> Phasor Measurement Unit description </swe:Discussion>
</description>

<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- Identification of the Sensor used    -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<identification>
   <IdentifierList>
      <identifier name="longName">
         <Term qualifier="urn:ogc:def:identifier:longName"/>
      </identifier>
   </IdentifierList>
</identification>

<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- What is the classification of Sensor-->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<classification>
   <ClassifierList>
      <classifier name="intendedApplication">
         <Term qualifier="urn:ogc:def:classifier:application">Sub-Station</Term>
      </classifier>
      <classifier name="sensorType">
         <Term qualifier="urn:ogc:def:classifier:sensorType">transformer</Term>
      </classifier>
   </ClassifierList>
</classification>

<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- When is these paremeteres defined valid -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<validTime>
   <StartTime>2007-12-01</StartTime>
   <EndTime>currentTime</EndTime>
</validTime>

<!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<!-- Who to contact -->
<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
<contact role="urn:ogc:def:identifier:operator">
   <ResponsibleParty>
      <organizationName>Mississippi State Univesity</organizationName>
      <contactInfo>
```

```
          <phone>
            <voice>662-325-2323</voice>
            <facsimile>n/a</facsimile>
          </phone>
          <address>
            <deliveryPoint>Mississippi State</deliveryPoint>
            <city>Starkville</city>
            <administrativeArea>MS</administrativeArea>
            <postalCode>39762</postalCode>
            <country>USA</country>
          </address>
        </contactInfo>
      </ResponsibleParty>
    </contact>

    <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <!-- Documents-->
    <!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <documentation role="documents">
      <DocumentList>
        <member name="website">
          <Document>
            <description>
              <swe:Discussion>Mississippi State University, ECE</swe:Discussion>
            </description>
            <fileLocation xlink:href="htp://www.ece.msstate.edu"/>
          </Document>
        </member>
      </DocumentList>
    </documentation>

    <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <!-- Reference Frame of PMU -->
    <!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
    <referenceFrame>
      <gml:EngineeringCRS gml:id="STATION_FRAME">
        <gml:srsName>Weather Station Spatial Frame</gml:srsName>
        <gml:usesCS xlink:href="urn:ogc:def:cs:xyzFrame"/>
        <gml:usesEngineeringDatum>
          <gml:EngineeringDatum gml:id="STATION_DATUM">
            <gml:datumName> Spatial Datum</gml:datumName>
            <gml:anchorPoint>origin is at the base of the mounting. Z is along the
axis
              of the mounting pole - typically vertical. X and Y are orthogonal to Z,
              along the short and long edges of the case
            respectively.</gml:anchorPoint>
```

```xml
        </gml:EngineeringDatum>
      </gml:usesEngineeringDatum>
    </gml:EngineeringCRS>
</referenceFrame>

<!--================-->
<!-- Sensor Inputs -->
<!--================-->
<inputs>
    <InputList>
      <input name="current">
        <swe:Quantity definition="urn:ogc:def:phenomenon:current"
          uom="urn:ogc:def:unit:ampere"/>
      </input>
      <input name="voltage">
        <swe:Quantity definition="urn:ogc:def:phenomenon:voltage"
          uom="urn:ogc:def:unit:volts"/>
      </input>
    </InputList>
</inputs>

<!--=====================-->
<!--Sensor Outputs-->
<!--=====================-->
<outputs>
    <OutputList>
      <output name="Measurements">
        <swe:DataGroup>
          <swe:component name="time">
            <swe:Time definition="urn:ogc:def:phenomenon:time"
              uom="urn:ogc:def:unit:iso8601"/>
          </swe:component>
          <swe:component name="current">
            <swe:Quantity definition="urn:ogc:def:phenomenon:current"
              uom="urn:ogc:def:unit:ampere"/>
          </swe:component>
          <swe:component name="currentPhase">
            <swe:Quantity definition="urn:ogc:def:phenomenon:currentPhase"
              uom="urn:ogc:def:unit:degree"/>
          </swe:component>
          <swe:component name="voltage">
            <swe:Quantity definition="urn:ogc:def:phenomenon:voltage"
              uom="urn:ogc:def:unit:volts"/>
          </swe:component>
          <swe:component name="voltagePhase">
            <swe:Quantity definition="urn:ogc:def:phenomenon:voltagePhase"
```

```
                    uom="urn:ogc:def:unit:degree"/>
                </swe:component>
              </swe:DataGroup>
            </output>
          </OutputList>
        </outputs>

        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <!-- Parameters of PMU -->
        <!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <parameters>
          <ParameterList>

            <parameter name="CurrentTransformer">
              <swe:Count>1</swe:Count>
            </parameter>
            <parameter name="VoltageTransformer">
              <swe:Count>0</swe:Count>
            </parameter>
          </ParameterList>
        </parameters>

        <!--~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <!-- Processes list of PMU -->
        <!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-->
        <processes>
          <ProcessList>
            <process name="clock" xlink:href="urn:GPSClock:1.0:001"/>
        </processes>

      </System>
    </SensorML>
```

APPENDIX B

SPARQL QUERY

APPENDIX B

SPARQL QUERY

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cim:<http://iec.ch/TC57/2007/CIM-schema-cim12#>
SELECT ?TerminalID ?timeStamp ?Value ?MeasurementType FROM
<http://www.ece.msstate.edu/~website /Voltage-Status.xml>
WHERE
{
 ?s cim:Measurement.Terminal ?TerminalID.
 ?s cim:MeasurementValue.timeStamp ?timeStamp.
 ?s cim:AnalogValue.value ?Value.
 ?s cim:Measurement.MeasurementType ?blankNodeID.
 ?blankNodeID cim:IdentifiedObject.name ?MeasurementType.
}
```