Sensor Agnostic and Communication Independent SCADA

by

Gustavo Ariel Naigeboren

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Sensor Agnostic and Communication Independent SCADA

By

Gustavo Ariel Naigeboren

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

Abstract

SCADA solutions have typically been reserved for large organizations because of the high costs involved in customization, including the sensors, the communication channels and the specific implementation. In recent years with the evolution of a wide range of technologies there is an opportunity for a variety of both large and small organizations to utilize SCADA systems for monitoring and control. The development of new and affordable technologies in many areas such as computers, sensors and networks will help to achieve the goal of low cost and more universal SCADA systems. The objective of this thesis is to create prototype SCADA framework as independent as possible of the sensors or communication channels used, keeping in mind affordability for small or medium size organizations. Through the process of developing SCADA systems across various application domains, experience is gained and recommendations generated toward developing a universal SCADA framework that will be largely independent of sensors and underlying communication technologies.

Acknowledgements

I want to thank every person who helped me with ideas, suggestions and support to complete this research work. One of the most important things I learned through the last four years of study is that it is impossible to do a thesis like this one without the help of a lot of people.

I would like to start to thank Dr. Robert McLeod for his continued support, great ideas and the way he encouraged me to achieve my goals. Also, I would like to thank Dr. Marcia Friesen for her great spirit, knowledge and for helping me to be a better person.

The sandbag project could not be a reality without the unconditional help and support of Dr. James Blatz. He gave me the freedom to design the best possible solution to monitor sandbag dikes. I want also to thank Brian Oleson for letting me use his house to test the system. Moreover, I want to thank him not only for having the patience to be there every morning at 7.30 am but also for all the coffee and breakfasts he made for me.

The idea to create a system to track patients in the nursing facilities came up after a meeting I had with Sandra Delorme at Saul and Claribel Simkin Centre. She was the first one who introduced me to the world of RTLS. There are a lot of people that helped me to understand the need and the possible solutions. I will also like to thank several people from WRHA - Winnipeg Regional Health Authority - who trusted me and helped me to understand how to create a solution for their needs.

There is one special thank you that I want to make to my past employers: Infomagnetics Technologies and Conviron for supporting me during my studies in the Master's program.

Also, I want to thanks my friends for their unconditional help, encouragement and support.

Finally, I want to thanks my wife and kids for their patience and encouragement they have with me.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Purpose

SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to collect data often remotely, log, and process it and to take action if needed. As the name indicates, it is not a full control system, but rather focuses on the supervisory level [1]. Usually, a SCADA system is composed at least one computer system with a communication interface that connects the system to the remote unit to gather data or to control it. A simple example could be a weather station installed outside a building. A computer is connected to the external system to collect data. If the temperature drops below a threshold, the computer will turn the heating system on. This system described above can be easily replaced by a temperature control system. The point is that there is typically some measurement, remote control and actuation that is computer controlled.

There is an analogy between SCADA systems and philosophy. There are many definitions of philosophy. The one used by Aristotle is that the philosophy is all, existing knowledge. After a particular field gets specialization, it will create a new thread such as theology or math [2]. This idea also applies to SCADA systems. SCADA is the generic term used to describe systems with remote sensors and logging capabilities. For example, an Alarm System is a SCADA system but for an specific purpose. An alarm has sensors to collect data. If an intruder breaks into a building and the system is active, it will alert the police and trigger an alarm. In other words, the system collects the data and based on some state it will take action. This action can also be manual. When a system gets to certain level of specialization, it will use a name to indentify it for that specific purpose. Other examples could be smoke or fire detection systems or HVAC - Heating Ventilating Air Conditioning - control systems.

SCADA has been typically used only by large organizations or manufacturers to control complex systems and processes. Recently, there has been a significant need and opportunity to monitor and control smaller systems. Two such systems discussed here are a sandbag monitoring system and a RTLS - Real Time Location System - to locate people or equipment indoors.

High end SCADA systems are not a suitable fit for small applications. Over the last ten years the author has been developing SCADA solutions for different kinds of companies such as a utility company, an environment control company and for the Department of National Defence (Canada). The need to create a common architecture to offer SCADA solutions at a lower cost was evident, independent of the sensors or the communication channels used to connect the central station with the RTUs - Remote Terminal Unit - as well as the operator or user.

One of the difficulties noted was the lack of a standards track for developing SCADA applications. Although an official IEEE SCADA standard was evolving it was not widely used in practice as the majority of SCADA systems were built "one-offs". The high degree of specialization was a consequence of the application itself as well as the myriad of sensors across various SCADA systems.

The intention of the present research project is to gain understanding of the potential to create a SCADA framework that is sensor and communication protocol agnostic. Specifically, the goal of the investigation is to see if it is possible to create a reusable architecture to create a SCADA framework independent of the sensors and communication channels to the largest degree possible. All SCADA systems have some degree of customization by definition. The goal here is to minimize that level of customization on behalf of the application developer, thereby, facilitating reuse and redeployment of the basic SCADA framework.

SCADA has not evolved in the same way that internet applications have. SCADA does not have an open architecture like modern software applications. This project will present alternatives to make the software part of SCADA agnostic of the sensors and protocols used by the sensors. There is no reason why a software engineer (SCADA application developer) should know the details of SCADA that are not related to the field of interest. As an analogy, the internet application developer does not really need to know the details of the transport layer protocols to develop web applications. The work done suggests an alternative architecture that could evolve to become a simple solution for many SCADA applications with emphasis on ease of implementation and cost-effectiveness.

The data collected by SCADA often has a lot of uncertainty and is statistical in nature. As a consequence, additional analysis may be required. Although, it is not part of a SCADA framework to provide analysis results, the architecture suggested here will facilitate the integration of these types of modules.

In order to achieve the thesis goals, two different applications were prototyped although with the view that the thesis concerns the systems and not the specific applications. Chapter 3 is a SCADA solution to monitor sandbag dikes and Chapter 4 is a SCADA solution for an RTLS - Real Time Location System - application. Chapter 5 presents a potential combination of the SCADA solutions presented in chapters 3 and 4 as another SCADA solution reusing what was created previously. Chapter 6 contains the final conclusions and future work developing a more universal SCADA framework.

Most SCADA systems, historically and now, have been non-standardized, "one-off", customized systems for specific applications. This reality is partly due to the non-uniformity of applications,

the fact that many SCADA systems are connecting to legacy systems, and that the components are numerous and diverse. The SCADA systems that are prototyped in this work can shed light on what should be kept and integrated into SCADA standards or a standardized SCADA architecture.

This work derives its value from three components. First, SCADA systems are evolving and this work provides insights into a potential evolving standard for SCADA systems, including insights into which parts are best to standardize. Second, it demonstrates that SCADA systems can add value by, for example, connecting to statistical engines that add a measure of intelligence to the system. Third, the SCADA systems then become highly suitable to applications where data has uncertainty or stochastic behaviour associated with it. Many of the emerging applications for SCADA systems, such as environmental monitoring, have this characteristic.

This thesis sheds light on which SCADA technologies and modalities will survive, and is a step towards demonstrating what a universal SCADA framework would encompass.

1.2 Scope

This thesis is divided in five different sections:

Chapter One: This chapter presents an introduction to the SCADA technology. Also, it details the objectives and goals of the research work.

Chapter Two: This chapter presents the best practices to develop SCADA systems. This chapter has two main sources. The first is the IEEE SCADA standard. The second source is my personal experience with more than 10 years developing SCADA solutions.

Chapter Three: This chapter presents a real implementation of a SCADA system used to monitor sandbag dikes for potential structural damage or movement during a flood.

Chapter Four: This chapter presents a real implementation of SCADA system to create a RTLS to monitor people, equipment or inventory.

Chapter Five: This chapter presents a potential evolution/combination of the SCADA projects presented in chapters three and four.

Chapter Six: This chapter presents the final conclusions and future work.

Chapter 2: SCADA Standard Practices

2.1 Definition

SCADA is an acronym for Supervisory Control and Data Acquisition. The main objective of SCADA is to do remote monitoring and/or control of remote devices over a communication channel and present the data to the user in friendly manner. As the name indicates, it is not a full control system, but rather focuses on the supervisory level [2]. Another possible feature of SCADA is the capability to not only acquire data but also to record it for future analysis. Furthermore, SCADA systems can also log information about the status of the remote equipment [3].

A SCADA system is composed of at least one central computer system called a master station (MS) that sends or receives the requests to one or more remote stations to get data (acquisition) or to execute an action (control). The term to identify the user interface is usually called HMI (Human Machine Interface), but in this work it is referred to GUI (Graphic User Interface) to promote SCADA as a software application rather than part of the hardware solution. Another component of SCADA is one or more field data interface devices - usually RTUs (Remote Terminal Unit) or PLCs (Programmable Logic Controller)- to interface with the monitoring devices and control systems. The last element is the communication channel used to transfer data between the remote device and the master station [4].

Much of this chapter is an adaptation of the SCADA IEEE standard (used with permission from IEEE std. C37.1-1994 (Superseded) IEEE Standard Definition, Specification and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control. Copyright 1994, by IEEE. IEEE disclaims any responsibility or liability resulting from the placement and

use in the described maner), notes, articles and personal experience building SCADA solutions over the years.

2.2 Introduction

Actual SCADA systems are generally limited to large organizations. One of the main reasons is the excessive costs of SCADA solutions limiting its use to a small group of companies. The present chapter is based on the IEEE SCADA standard which is demonstrably complex, and as such has been simplified here updating the technologies used. The range of current SCADA systems is complex and disorganized. This work sheds light on how a standardized SCADA architecture could evolve.

There are many different implementations of SCADA from different vendors. There is a need to create a unique SCADA framework to reduce costs and to create a unique platform that is agnostic of the sensors used or the remote devices controlled. There is also a need to have SCADA frameworks that handle multiple protocols to communicate with the remote devices.

2.3 System Functionality

The SCADA system is composed of three basic elements as shown in Figure 2.1.

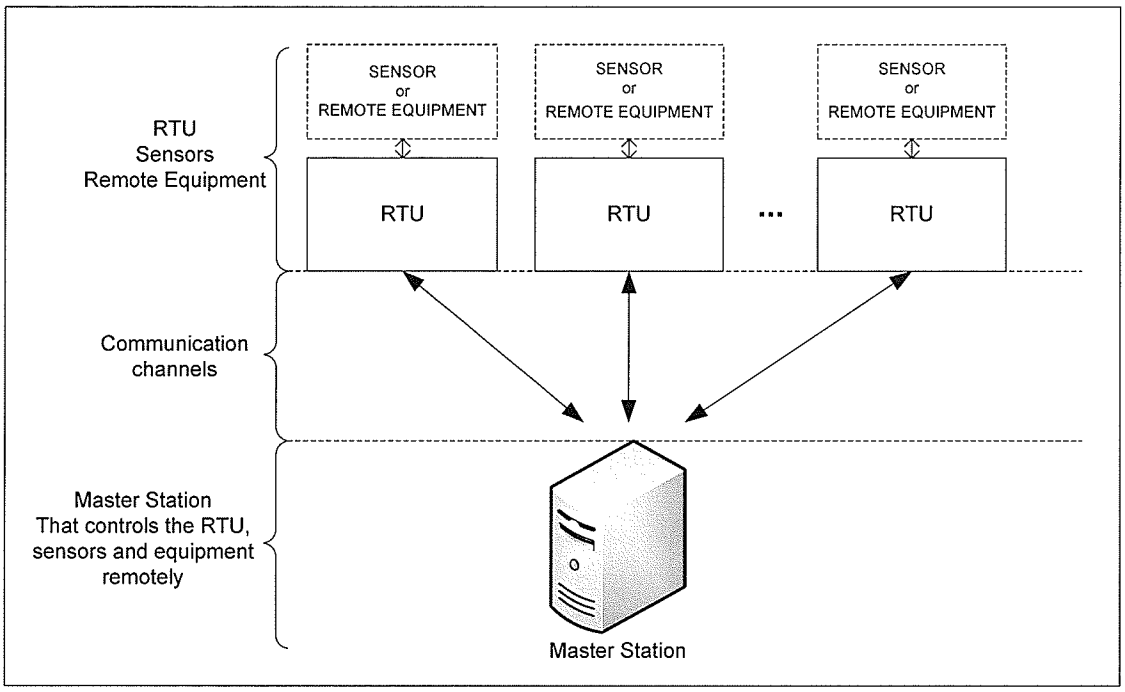

Figure 2.1: Typical SCADA system

There is no unique configuration for a SCADA system. However, a typical system will have at least one master station (MS) and at least one remote terminal unit (RTU). Usually, the protocol used requires that the MS initializes the communication with the RTU. The MS is the entity responsible for polling all the RTUs for updates of status and sensor information (if a polling protocol is used) [3].

There are two very distinctive modes to get data from the RTUs using SCADA. The first type is where the master station polls the RTUs to request data from the sensors. The second type is where the master station received that data from the sensors without requests (interrupt mode). In Chapter 3 an example of how to build an SCADA system to monitor a sandbag dike for potential

structural damage is prototyped using the polling technique. Chapter 4 presents another example of how to build an SCADA for RTLS is prototyped using the interrupt mode. It is also possible to utilize a combination of both systems. There is a big challenge to use interrupt mode using wireless technology because of collisions in the transmission process. This could delay an alarm to be sent to a master station. I had a challenge in a system deployed for a hydro company in Argentina - with over 200 sensors - where I had situations where the remote devices were experiencing significant delays getting access to the wireless channel while attempting send data back to the master station. The network used was a low speed connection, however this problem would be present on a high speed network if sufficient sensors or significant amounts of data were being collected.

The communication channel is the media that connects the remote equipment or RTU with the master station. There are two main groups: wired and wireless. The wired technologies are more commonly used in areas where the element to be monitored or controlled is close to the master station. This solution is also used in fields like telecommunications where all the nodes are connected with an existing network. The second alternative - wireless - has grown significantly in recent years. One example is shown in Chapter 3 where the sensors in the sandbag dike are connected using a wireless mesh network. Another possibility is to use the cell phone companies as a transport solution for the data.

2.4 System Configurations

The master station can be composed of one or more computer systems. The computer systems can be centralized or distributed over different locations (Figure 2.2). Also, the system can have multiple communication modules.

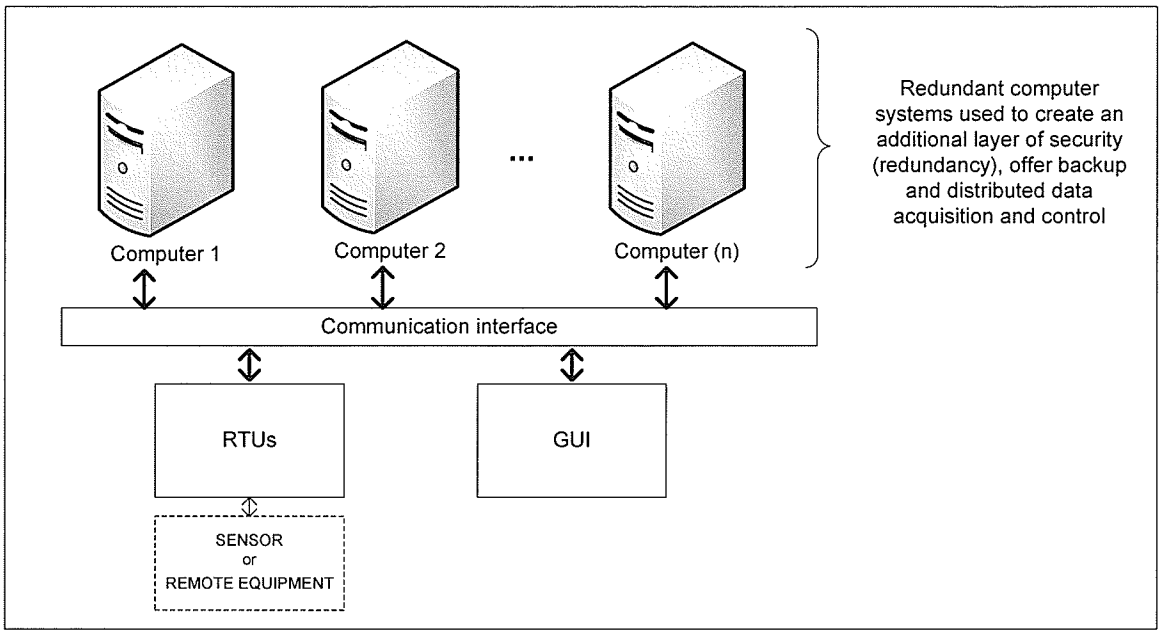

Figure 2.2: Example of redundant and distributed or centralized SCADA

There are many reasons to install a distributed computer system or multiple communication modules (different communication links between nodes). Some of the reasons are:

a) System redundancy: in case of failure there is another system as a backup.

b) Multiple plant building: a plant can have multiple buildings with one SCADA per building.

c) Different tasks: one computer system can be acquiring the data while other is recording the information into a database and retrieving it and presenting it to the end user.

d) Communication channel saturation: because of physical limitations on the communication channel there might be a need to have multiple master communication devices to contact the RTU. For example, if a system is using a wireless communication channel there is a limit of how many RTUs can be installed to guarantee them access to the MS in time to report data or alarms. If a channel is saturated, some RTUs might not find a timeframe to send data back to a MS.

2.5 System Features

The SCADA system can have multiple features or modules. SCADA systems differ from each other based on requirements. The following is a list with the most important components to consider in a SCADA system.

a) User Interface

The user interface or HMI (Human Machine Interface) or GUI (Graphic User Interface) is the interface between the system and the user. It has to be user-friendly and intuitive. The critical indicators have to be visible. Alarms and status indicators have to be clearly displayed on screen.

The goal of the user interface is obvious. However, there have been accidents caused by poorly designed systems or missing alarms. On February 14, 1982 the drill rig Ocean Ranger sunk into North Atlantic waters because operators could not understand the extent of a problem with the platform due to a poorly designed GUI. Also, the lack of electronic alarms failed to notify the crew of a potential problem. Another example is the accident at Three Mile Island on March 28th, 1979. The nuclear plant was only thirty minutes away from the meltdown. What saved the plant was an experienced operator who was called at 4.00am. The system was not capable of notifying the operator on duty of the problem [5].

b) Data Processing

The SCADA system's principal data processing task is to acquire data from a RTU. Then the system can optionally process the data before showing it to the user or storing it into a database.

c) Database Maintenance

The database is a key component of the SCADA systems capable of recording information for future analysis or reference. The database has to be fully functional and properly maintained to avoid critical failures. A typical problem of SCADA systems is database growth, filling up the storage device making the whole system crash or become inoperable.

d) Control Processing

The control processing is another optional feature for SCADA systems. For example, a SCADA can be logging data from a remote sensor and can take action in case of an alarm, major failure or just control a process. The 'control' side of the SCADA can be automatic or manual assisted by an operator.

e) System Backup

Usually, a SCADA system requires a very intensive configuration to function properly. Also, the historical data retrieved by the system is stored in a database. Special care is required at the time of the backups not only to restore a system in case of failure but also because some of the logged data may have to be kept following regulatory rules.

f) System Redundancy

SCADA systems are sometimes used in very critical situations. They can be as simple as taking temperature values from an outdoor sensor, or they can be responsible to manage the whole electrical distribution of a region. System redundancy is sometimes required. The systems can be located in different buildings in some cases. Also, there are cases where the RTUs are duplicated to have a backup in case of a failure of the main RTU device. Furthermore, there are cases where the communication channels are also duplicated in case of malfunction.

g) Self Diagnostics

The SCADA system is a compound of many subsystems. A self-test diagnosis is required to understand the functionally of each sub component. The system can be created in a way that the central system not only gathers valid data from sensor but also critical system information such as RTU internal temperature, external meteorological conditions or communication channel status.

h) Communication Interfaces

a. Communication with another computer system

Since SCADA systems can be distributed, it is critical to have a reliable system to interconnect the main computer systems. An example of this system could be a regional system that controls energy distribution. The networks can be LAN or WAN, and some combination of public and private.

Also, there are some SCADA configurations where many computers are needed. For example, you can have a system to communicate with the RTU, other one to record data into a database and a third one to present it to the user. The interconnection between the systems is usually done using a reliable LAN or WAN network.

b. Communication with a RTU device

The communication between the main computer system and the RTU can be done using LAN or WAN networks, serial connections, specific manufacturer's technologies, wireless, standard or proprietary.

i) Analog Inputs (usually connected to the RTUs)

Including but not limited to transducers and sensors.

j) Analog Outputs (usually connected to the RTUs)

Including but not limited to controllers, recorders and meters.

k) Digital Inputs (usually connected to the RTUs)

Including but not limited to pulse inputs, breakers, switches and relays.

l) Digital Outputs (usually connected to the RTUs)

Including but not limited to breakers, switches, generators, other devices.

2.6 Additional System Definitions

There are several elements that have been defined for a SCADA system, most of which are self explanatory.

2.6.1 Communication Management [3]

a) Message protocol

b) Number of communication channels

c) Channel considerations

d) Error detection techniques

e) Channel switching

f) Number of RTU per channel and/or channels per RTU

g) Number of retry attempts

h) Time out value(s) by message type

i) Communication error reporting, failure, criteria, and recovery

j) Channel quality monitoring (normal and backup)

k) Channel diagnosis/test provisions

l) Equipment interfaces

m) Report-by-exception of point scan

2.6.2 Data acquisition

The definition for each possible type has to be defined. All the possible ranges for data input and output, scale factors, rates, and accuracy also have to be defined [3].

Each RTU has to have a definition of its capacity. Typical elements are the number of inputs, number of outputs and data rate. The MS also has to have definitions to interface with the RTU properly such as data exchange rate with the RTU or communication media.

The RTUs can process data before sending it to the central system. If that is the case, exceptions have to be implemented to alert the MS of possible errors.

2.6.3 Data types

There are different kinds of data that can be retrieved from the RTU. The following list describes the most important types and the considerations required in each case.

a) Analog data: It is a very important to define where the analog data will be processed. The information about filters, amplifiers, single or multiple data reading processes has to be considered.

   The analog data sent to the controller can be one of the following:

   - Accumulated data
   - Computed data
   - Alarm data

b) Status Data: The information contained in a digital signal is based on the discrete states of the signal such as presence or absence of a voltage, current or contact.