DEVELOPING OF A HIERARCHICAL INTERFACE ON-LINE DATA FROM SCADA SYSTEMS TO ERP SYSTEMS

Marwan[1)]

Abstract: The purpose of this research is to address the problem of on-line data interconnectivity with development of a hierarchical interface which will be able to interchange on-line data from SCADA (Supervisory Control and Data Acquisition) systems to business process systems such as Enterprise Resource Planning (ERP) systems. Some preliminary investigation has been conducted such as the function of real-time data acquisition and monitoring systems of a water utility, and data filtering using a real-time compression algorithmic approach. In order to achieve the aim and objectives of this research, further modelling must be conducted such as refining the filtering algorithm, modelling systems and data hierarchy, identifying system related criteria and developing a hierarchical interface database.

I. INTRODUCTION

Most organizations use computer technology in their business activities. Computer technology has penetrated every facet of business activities, from simple calculators to complete ERP Systems. Efficient, fast and effective data acquisition is the hallmark of computer technology.

However, plaguing this computer technology are many problems relating to data management. The speed of collection of data far exceeds the speed with which organizations analyse it and draw useful information. This problem is exacerbated with on-line data acquisition systems, specifically data obtained from SCADA systems.

Condition monitoring of machines in engineering plays a very important role in their use, as it provides information on the operational characteristics, health and maintenance requirements of the machine. Unscheduled breakdowns of mechanical equipment in process plant can therefore be reduced. Condition monitoring is continually undergoing rapid development with new techniques in the areas of tribology, thermography, acoustics and vibration, in addition to basic monitoring of operational characteristics such as temperature, pressure, flow, etc.

Advances in computer technology have had a significant impact on this process. Advancement in sensor technologies, automated controls and also data telemetry has made possible new and innovative methods in condition monitoring. Rapid growth in networking systems especially through the Internet have overcome

[1] Staf Pengajar Jurusan Teknik Mesin Politeknik Negeri Ujung Pandang

the barriers of distance, allowing, real-time data transfer to occur easily from different locations. With real time data transfer, current equipment condition can be measured immediately and any operational abnormality easily identified.

A specialized system for real-time condition monitoring, namely SCADA, superceded Distributed Control Systems (DCS) in industrial process plant. This system is basically the state of the art of real-time monitoring systems. Based on its name, the duties of this system are supervisory control and data acquisition.

However, analysing data obtained from a SCADA system is complicated and difficult. Usually any abnormality from a monitored system beyond specified set points of operation triggers an alarm which is either recorded, or recorded together with a physical alarm. This is based on the setting of specific alarm limits in a particular operational characteristic (i.e. temperature, pressure, etc.) of the monitored system. However it is not easy to conduct post incident analysis to determine the root cause of any abnormality based on the produced data. Data mining research has attempted to address this problem through relational database techniques. Attempts at solving this problem through interconnecting data are not always effective, resulting in inaccurate analysis results (which is often followed by incorrect action). This is in most cases due to the fact that data from a SCADA system is nested data (data that is influenced by other data). Without an appropriate method for analysis, this process will be like entering a labyrinth in which many of the ways out are false.

This research will attempt to solve the problem by converging analyses not only to the data but also to the system. Thus, a monitored systems hierarchy will be modelled and analysed in order to structure the acquired data as well as the monitoring system, according to a logical hierarchy whereby systems related criteria can be identified.

A further problem in using data from a SCADA system is its difficulty to interconnect the data to other business process systems such as ERP systems. ERP systems include the software tools that are used to manage enterprise data such as a logical business supply chain, receiving, inventory management, customer order management, production planning, shipping, accounting, human resource management, and other business functions. Basically, ERP systems collect data from different organizational functions or departments and process and distribute the data to departments where they are needed. Data processed in ERP systems is predominantly about the movement of products or raw material, the movement of people and the movement of cash within a company. Until now, according to the preliminary literature review, there is no concept or commercial product that provides a direct hierarchical interface which is able to interconnect these systems. This research will address the interconnectivity problem by developing the concept of a hierarchical interface to interconnect SCADA and business process systems such as ERP systems.

The aim of this research is to determine through systems hierarchical modelling the feasibility of convergence and interconnectivity of on-line data acquisition systems with off-line business process systems. This will be investigated through the development of a hierarchical interface which will be able to interconnect SCADA with business process systems. However, there are several objectives that must be achieved to test the set hypotheses, prior to reaching a conclusion on the aim of this research. These objectives are:

a. Development of a data filtering algorithm, using the algorithmic approach of data compression, to remove redundant data.
b. Modelling of systems and data hierarchy of a case study scenario.
c. Development of a hierarchical interface database together with an on-line SCADA system interface data bridge for validation of the research findings.

II. RESEARCH METHODS

The first experimental method in this research is to test the research hypothesis given in said above, specifically on-line data acquisition systems such as SCADA can be converged through systems hierarchical modelling whereby systems related criteria can be identified. This research method uses random assignment of subjects and the manipulation of variables in order to determine cause and effect through the development of a simulated SCADA system in LabVIEW. The next step is simulate SCADA system in LabVIEW of Queensland University of technology Australia. The last step is prototype of an on-line data acquisition by monitoring system.

III. DISCUSSION

The study was initialised by designing system architecture that simulated a remote monitoring system for water quality, followed by prototype development and testing of an on-line data acquisition system.

The system structure is illustrated in Figure 1 which consists of three stations, namely; local measurement, central control, and a remote browse station. The local station is placed close to the measurement point, with the duties of measurement of water quality, displaying real-time data chart, basic data analysis, and data transmission to the central station, with response commands from the central control station.

All data from different local measurement stations are centralized at the central control station for further data analysis. Other duties of the central control station are to display the real-time water quality condition of different monitoring stations, store the data history, send commands and messages to the local measurement stations, and generate periodic reports of water quality. The main program on the central control station is used to analyse the collected data (advanced analyses), show diagrams of the measured parameters, and communicate with the measurement stations and remote browse stations. The remote browse station is a general computer used to log onto the web site of the water quality monitoring system to browse the information and retrieve data related to water quality.

The whole system is built over the Internet based on a client-server mechanism. The HTTP server of the central control station services the remote browse stations. It connects and interacts with a database through CGK (Common Gateway Interface) and/or ADO (ActiveX Data Objects) plus ODBC (Open Database Connectivity). The remote browser searches through retrieved data from the central control station or local measurement stations through an ActiveX control enabled Web browser, Java Applet, or HTML pages.

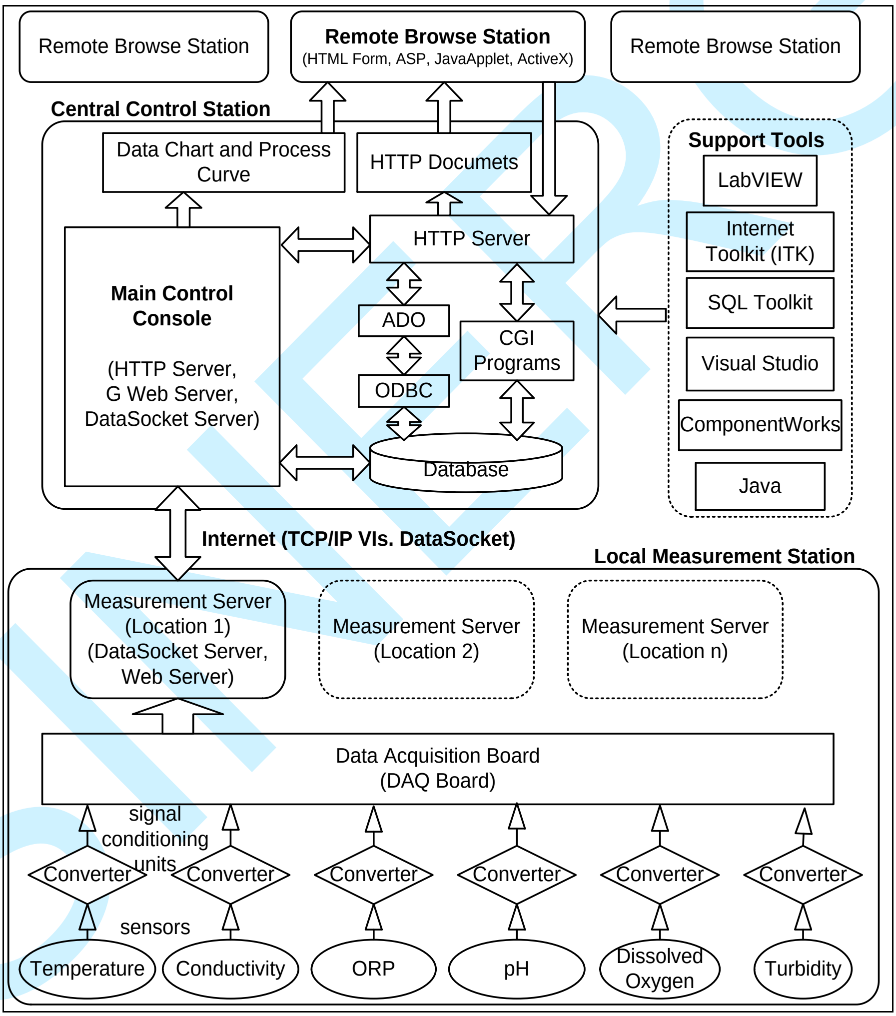


Figure 1. Layered Structure Diagram Of Remote Water Quality Monitoring System

The on-line data acquisition prototype was developed using LabVIEW 7.1 programming language platform. Data communication and transfer between the measurement stations and the central station across the network was implemented with LabVIEW's DataSocket Technology. The LabVIEW inbuilt function or developed application program within the LabVIEW environment is called VI (Virtual Instrument).

LabVIEW has provided a series of schemes for communication over the Internet [92] shown in Figure 2 These Web Technologies can support users by implementing remote monitoring, remote control, collaboration, and distributed computing.

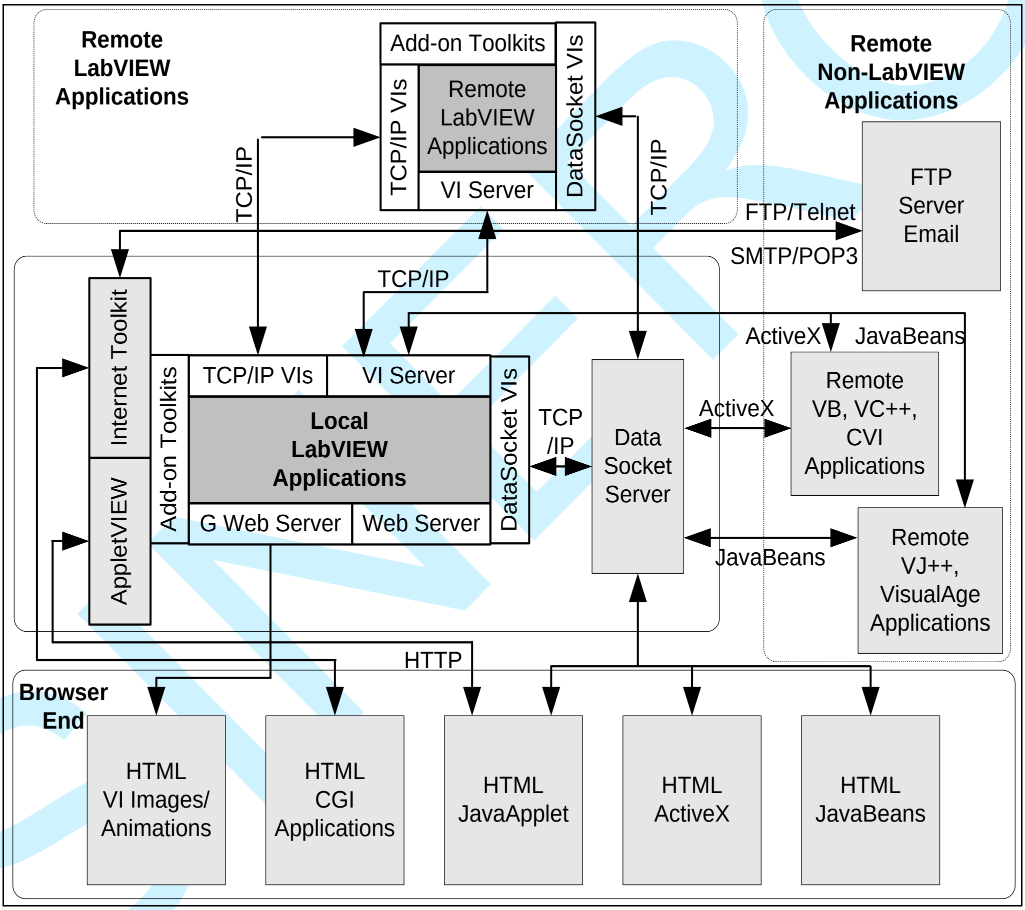


Figure 2. LabVIEW Based Web Technologies For Communication Over The Internet

Figure 3. shows a framework of the local measurement station consisting of sensors, signal conditioning units, a data acquisition board, and a measurement server (computer). The sensors collect data of temperature, electrical conductivity, pH, ORP, dissolved oxygen, and turbidity. Furthermore, the signals from the sensors must be converted into a suitable form for DAQ processing since most of the signals cannot be imported to the DAQ board directly.

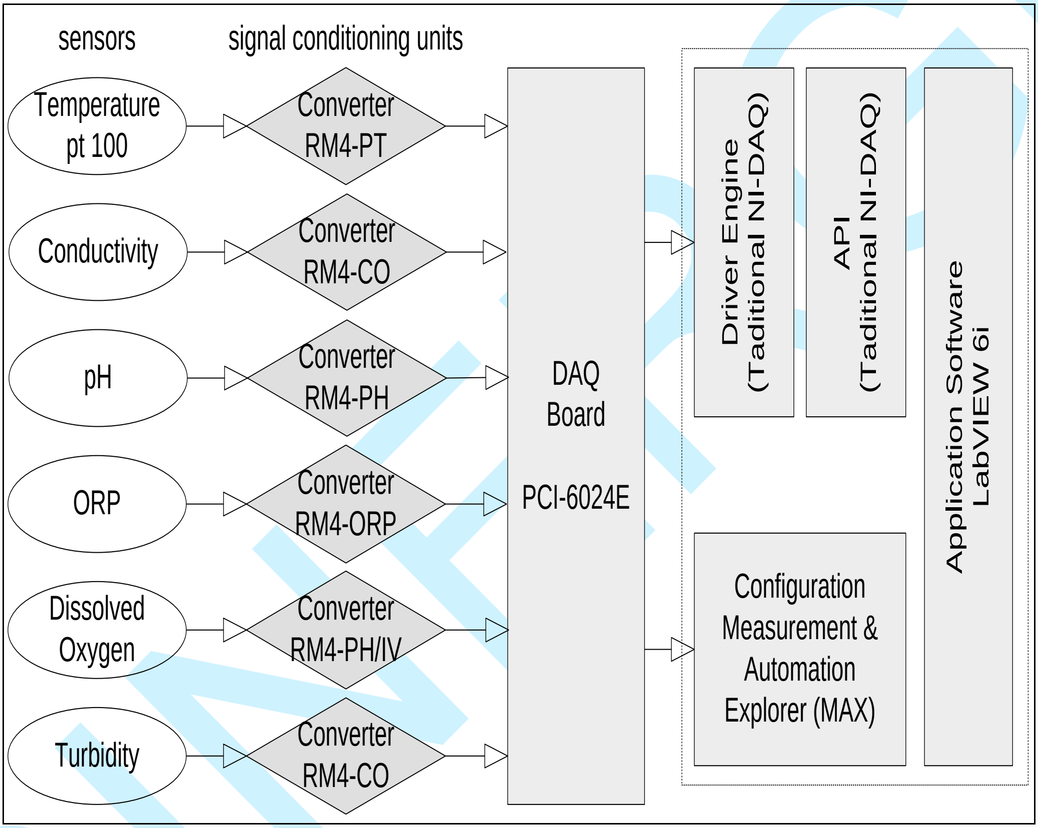


Figure 3. The Framework Diagram Of Local Measurement Station

The duties of the measurement server in this system are to collect data from and send commands to the DAQ board, display water quality condition in real-time, store the local history data, and also send/receive data from the central control station. LabVIEW Professional Developer Suite 7.1 was employed to develop the system. Communication between the local measurement station and the central control station is built on the TCP/IP protocol over the Internet.

The signal acquisition devices and data channels were set using the Configuration Measurement and Automation Explorer (MAX) toolkit. The data acquisition task was implemented by LabVIEW's built-in DAQ, Figure 4. illustrates a display of the local measurement station terminal.

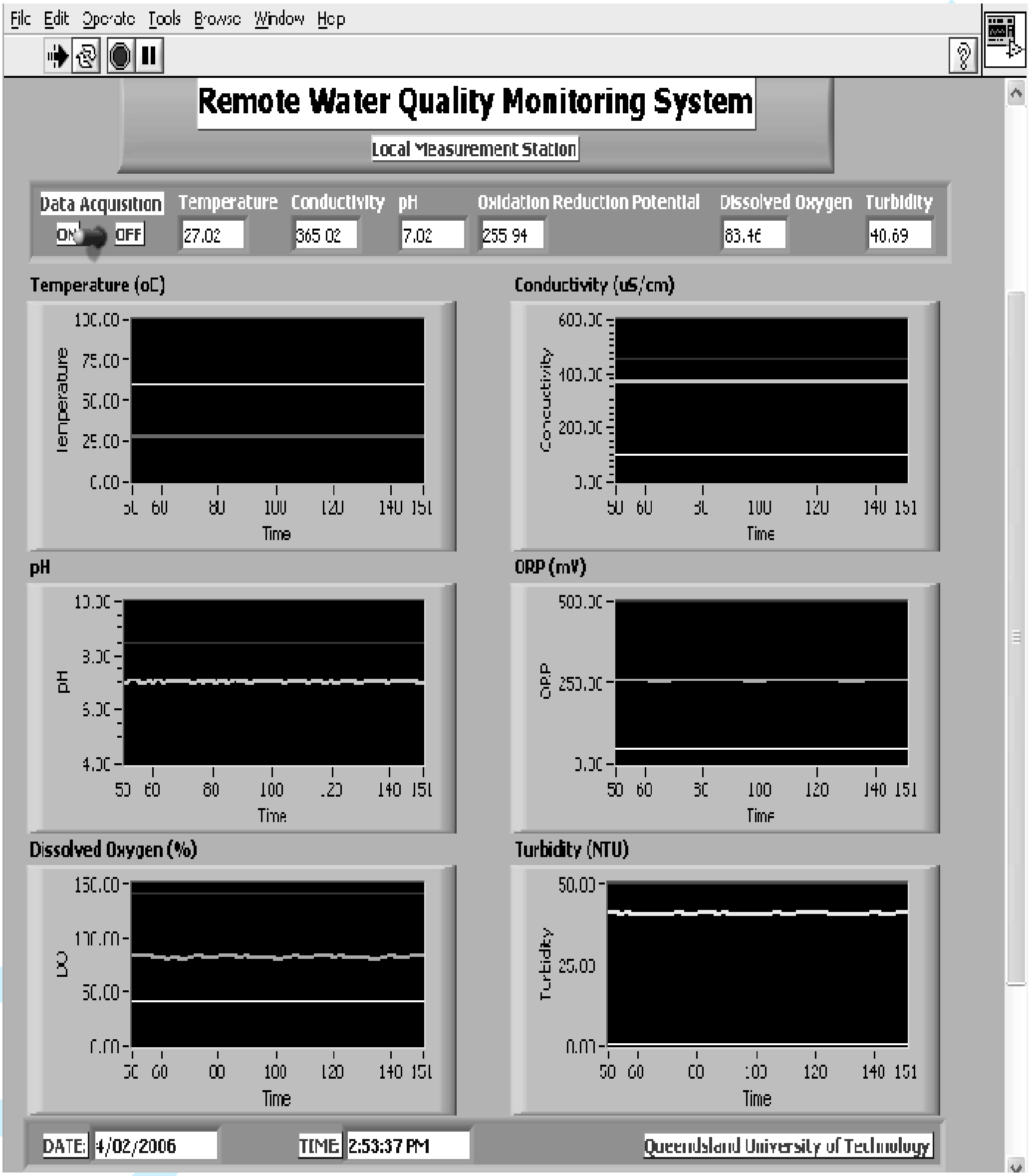


Figure 4. Real-Time Water Quality Condition Displayed On Local Measurement Station Terminal

The structure of the central control station is similarly shown in Figure 1 and its interface illustrated in Figure 5. As previously indicated, the duties of the central control station are to gather data from the local measurement stations, display the real-time water quality condition of the different monitoring locations through connection to the measurement station server, send commands and messages to the measurement stations, analyse the water quality condition through comparisons with the history record and the water standard specifications, store the history data, and generate periodic reports of water quality.

The communication between the local measurement station and the central control station is conducted through a DataSocket and/or TCP/IP VIs based on the HTTP protocol. The SQL database is used to store the whole data history. In the LabVIEW Professional Developer Suite 7.1 there is a toolkit (SQL toolkit) which is able to connect the SQL database and the application VIs. Some basic data analysis tasks can be done by using the LabVIEW built in functions.

In this prototype, there are three types of servers running on the central control station; a G Web Server, a Data Socket Server and a HTTP Server. The duty of the G Web Server is to link communication between the remote browse side and the LabVIEW application VIs of the server side through the CGI VIs. The DataSocket Server serves to build a connection and transfer data through the DataSocket to the measurement stations, the central control station, and the ActiveX embedded Web Browser (remote browse station). Finally, The HTTP server (IIS) operates as a main server of the central control station which is used to build the connection between a Visual Basic (VB) application database, and Web Browser. However, the port of the HTTP server must differ from the port of the G Web Server; otherwise, it will cause a server conflict.

Figure 5. An Interface Of Central Control Station Terminal

The remote browse stations are used to browse information in the server of the central control station and the local measurement stations. Internet Explorer and Netscape Navigator are suitable browsers as they are Active-X enable. All information from the central control station and the local measurement stations can thus be accessed by clicking on the Web Browser. In this study the ActiveX mechanism in VB is employed for communication between the Browse-side and Server-side. Figure 6. shows a web page on which a client at a remote browse station can review real-time water quality condition of a particular measurement station through the ActiveX-enabled Web Browser. The data history from the central control station and the local measurement stations can be downloaded by using FTP (File Transfer Protocol). Figure 7. illustrates the water quality data history.

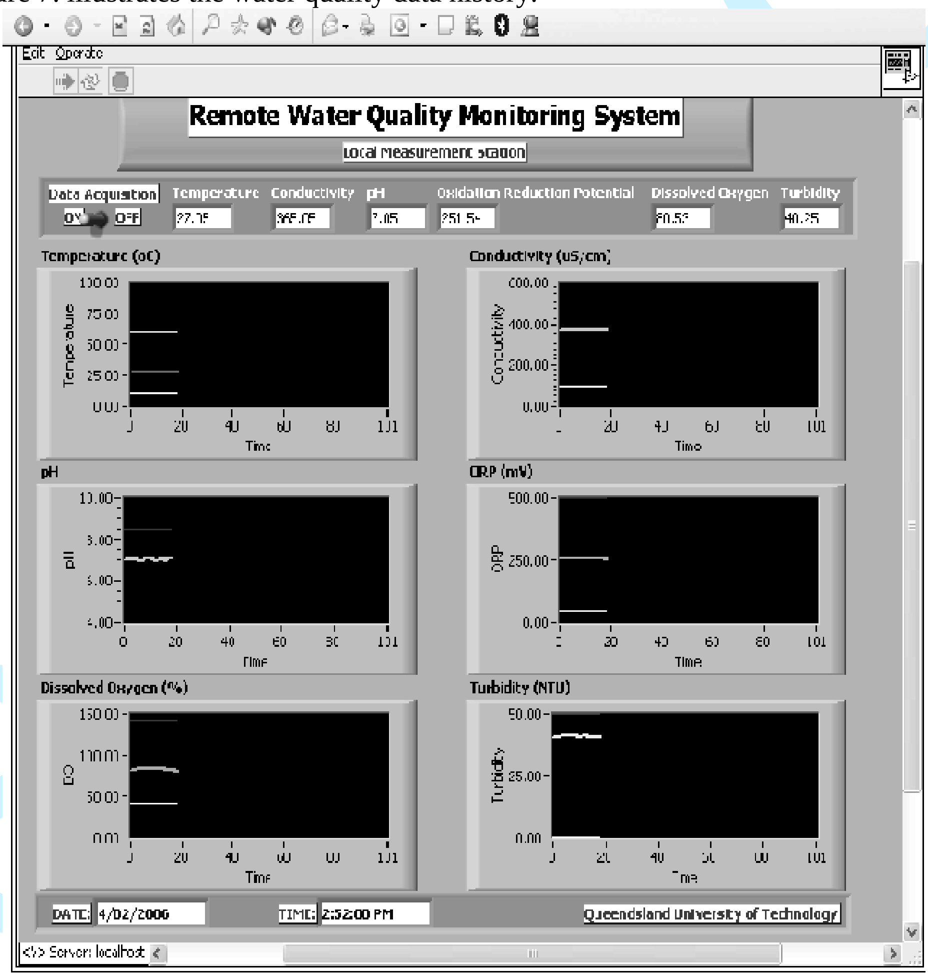
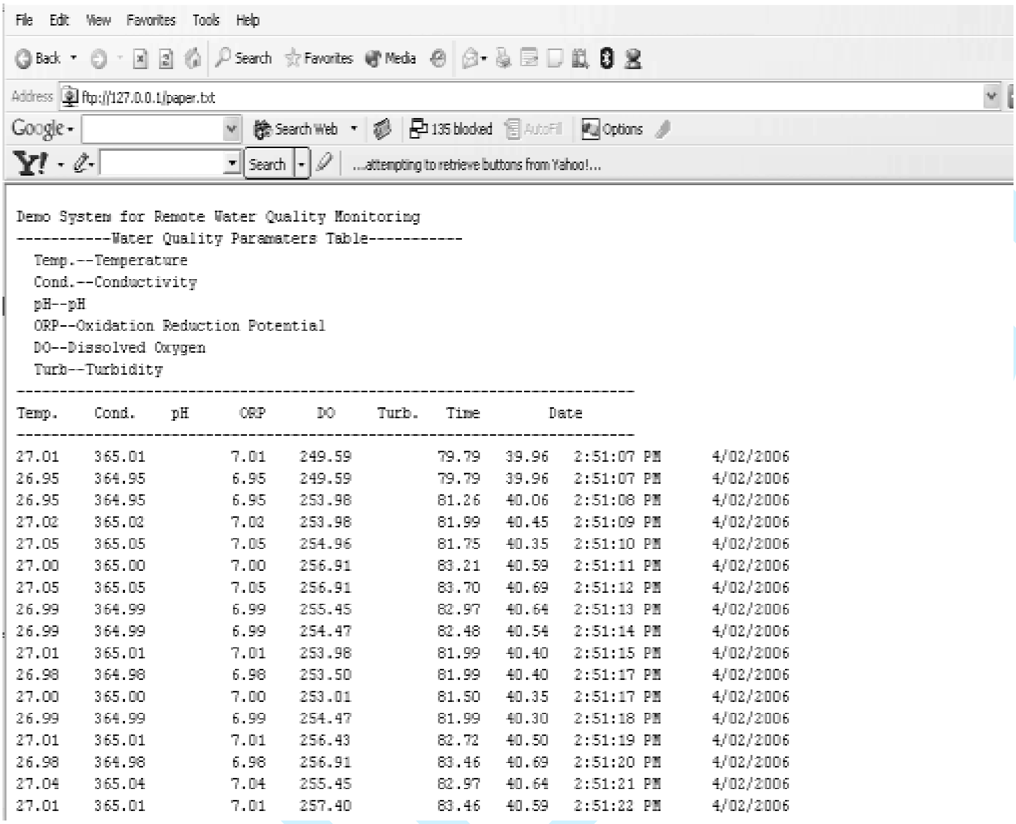


Figure 6. Water Quality Condition Displayed On The Remote Browse Station



Demo System for Remote Water Quality Monitoring
-----------Water Quality Parameters Table-----------
Temp.--Temperature
Cond.--Conductivity
pH--pH
ORP--Oxidation Reduction Potential
DO--Dissolved Oxygen
Turb--Turbidity

Temp.	Cond.	pH	ORP	DO	Turb.	Time	Date
27.01	365.01	7.01	249.59	79.79	39.96	2:51:07 PM	4/02/2006
26.95	364.95	6.95	249.59	79.79	39.96	2:51:07 PM	4/02/2006
26.95	364.95	6.95	253.98	81.26	40.06	2:51:08 PM	4/02/2006
27.02	365.02	7.02	253.98	81.99	40.45	2:51:09 PM	4/02/2006
27.05	365.05	7.05	254.96	81.75	40.35	2:51:10 PM	4/02/2006
27.00	365.00	7.00	256.91	83.21	40.59	2:51:11 PM	4/02/2006
27.05	365.05	7.05	256.91	83.70	40.69	2:51:12 PM	4/02/2006
26.99	364.99	6.99	255.45	82.97	40.64	2:51:13 PM	4/02/2006
26.99	364.99	6.99	254.47	82.48	40.54	2:51:14 PM	4/02/2006
27.01	365.01	7.01	253.98	81.99	40.40	2:51:15 PM	4/02/2006
26.98	364.98	6.98	253.50	81.99	40.40	2:51:17 PM	4/02/2006
27.00	365.00	7.00	253.01	81.50	40.35	2:51:17 PM	4/02/2006
26.99	364.99	6.99	254.47	81.99	40.30	2:51:18 PM	4/02/2006
27.01	365.01	7.01	256.43	82.72	40.50	2:51:19 PM	4/02/2006
26.98	364.98	6.98	256.91	83.46	40.69	2:51:20 PM	4/02/2006
27.04	365.04	7.04	255.45	82.97	40.64	2:51:21 PM	4/02/2006
27.01	365.01	7.01	257.40	83.46	40.59	2:51:22 PM	4/02/2006

Figure 7. History Data Of Water Quality Condition On The Local Measurement Station

The purpose of this study was to determine a means for filtering data that is produced from a SCADA system. The approach was to study the characteristics of the data from the SCADA system in detail. It was found that much data from a SCADA system is redundant in terms of relevance. Several experiments were conducted to observe the feasibility of reducing this redundancy, and to determine an effective means to filter unwanted data.

Data storage becomes a serious problem with on-line data acquisition systems since masses of data are collected and dumped into storage every day. Furthermore, if the storage is not managed properly it soon becomes overwhelmed by redundant data. For example, a single day measurement of 1 sample every 10 seconds, where the size of each sample is 8 bytes, requires 69.120 kilobytes storage. Ten channels of such data demand about 7 Mbytes of disk space. Hence redundant data filtering through compression techniques is important. In addition, if a large amount of data needs to be transferred to a remote station, the use of data compression greatly reduces the transfer time needed. Benedetti [94] proposed a method of compression by using the least significant word algorithm; however, it is not easy to implement this method.

Initially the data was stored and encrypted in the SCADA (Supervisory Control And Data Acquisition) archive file formats. With data reader software supplied by the SCADA provider, data was converted to a TXT format. It could then be manipulated (the manipulation process is only a simple mathematical process which is subtracting previous data from current data) to calculate a disparities pattern. The manipulation algorithm is given as:

$$Y(n) = d(n) - d(n\text{-}1) \quad \textbf{....} \ (1)$$

Y = data disparities
n = sampling number
d = data absolute value

Matlab 7.0.4 was employed to manipulate the data. The data that was used in this studies was obtained from a motor at a water utilities pump station. Originally there were 10 different parameters. However, initially only 2 parameters were used to generate the algorithm. These two parameters are temperature data from two different motor bearings.

Several types of data compression techniques have been proposed in different fields of application such as satellite communication, commercial telephony and image processing. Basically, these compression schemes try to encode the data so that the size of the data can be reduced. In real-time data acquisition and condition monitoring, there are several compression algorithms that have been proposed, for example Yeh developed real-time lossless data compression for remote sensing, and Chan developed real time data compression for power system monitoring. However, both of these algorithms are difficult to implement in SCADA systems.

The philosophy underlying the proposed algorithm in this study is that only the difference which contains new information is recorded. Thus, the lossless compression code is then generated to record the new information. In this way, data redundancy could be reduced.

The first step of the algorithm is to find out the disparity between the current data and the previous data. This disparity, which has 48 bits resolution, is coded to the one character with 8 bits resolution. Finally, the code from two different parameters is combined together which is represented by one character with 8 bits resolution (it is as though each of the parameters occupy 4 bits resolution). With this approach, the 4 bits compression will be easier implemented since the algorithm only manipulates the data string.

In this experiment, there are only 13 characters used since these characters are able to encode more than 99 % of the total disparities which is statistically sufficient in reducing redundancy. Adding more characters will not give a more significant result. The steps in the design of the algorithm are given in Figure 9 and 10. In conclusion, this algorithm is able to reduce data redundancy significantly.

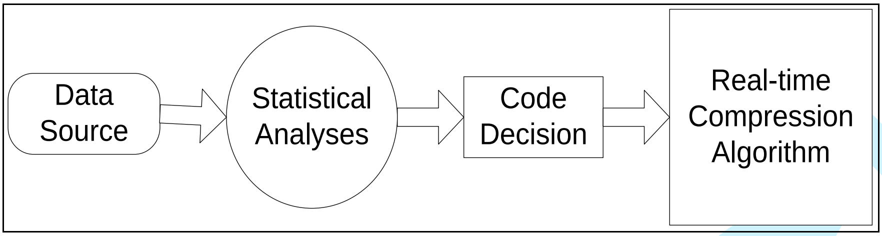


Figure 9. Algorithm Design

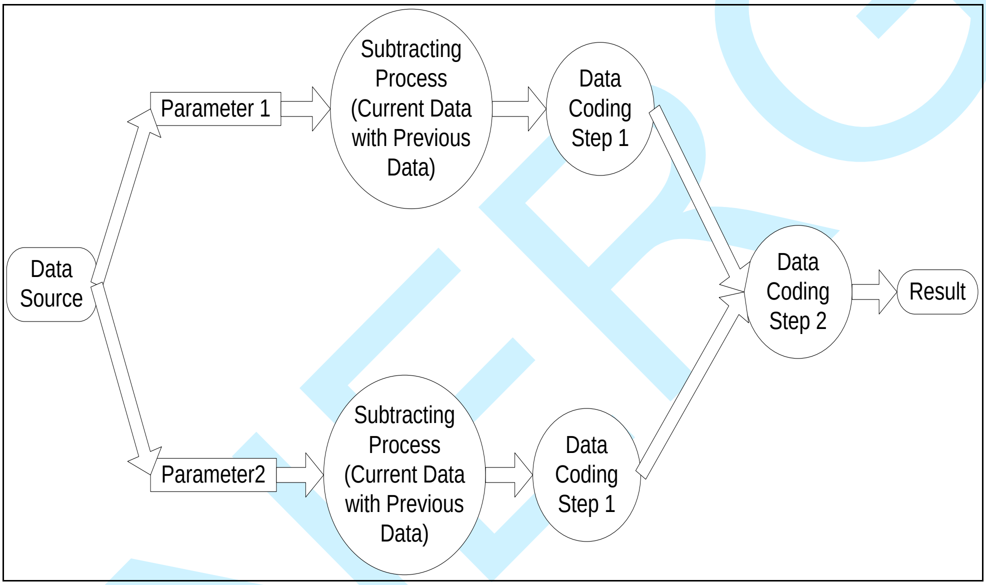


Figure 10. Real-time Compression Algorithm

IV. CONCLUSIONS

1. In the filtering process, SCADA data will be scanned with a selection algorithm to decide which data is relevant and which data is redundant. The relevant data will be held while the irrelevant data will be discarded. This filtering process will be used in the development of the on-line SCADA to hierarchical interface database.
2. An interface hierarchical database and related data hierarchy can be modelled, the simulated on-line data acquisition and monitoring system needs to be analysed. A system hierarchy can then be modelled based on this analysis. Once this has been done, the data hierarchy can be modelled, followed by

modelling of system related criteria. Finally, the interface hierarchical database can be modeled.

3. After successful completion of the previous steps, the results will be used and combined to formulate an on-line SCADA to hierarchical database data bridge. This data bridge will function as an interface to interconnect the SCADA system with the interface hierarchical database.

V. REFERENCES

A. Davies, *Handbook of condition monitoring : techniques and methodology*, 1st ed. London: Chapman & Hall, 1998.

A. G. Taylor, *Database development for dummies*. Foster City, CA: IDG Books Worldwide, 2001.

A. Laugier, N. Allahwerdi, J. Baudin, P. Gaffney, W. Grimson, T. Groth, and L. Schilders, "Remote instrument telemaintenance," *Computer Methods and Programs in Biomedicine*, vol. 50, pp. 187-194, 1996.

A. Matus-Castillejos and R. Jentzsch, "A time series data management framework," presented at Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, 2005.

A. Starkey, A. Ivanovic, R. D. Neilson, and A. A. Rodger, "Use of neural networks in the condition monitoring of ground anchorages," *Advances in Engineering Software*, vol. 34, pp. 753-761, 2003.

B. K. N. Rao, *Handbook of condition monitoring*, 1st ed. Oxford, UK: Elsevier Advanced Technology, 1996.

B. T. Kuhnell and J. S. Stecki, "New Wear and Contamination Sensors," *Condition Monitoring and Diagnostic Engineering Management*, vol. 2, pp. 543-551, 1998.

C.P.Ephraim, "Condition Monitoring In Automated Manufacture," *IEE Colloquium*, pp. 6/1-6/4, 1997.

F. B. Zhou, M. D. Duta, M. P. Henry, S. Baker, and C. Burton, "Remote condition monitoring for railway point machine," presented at Railroad Conference, 2002 ASME/IEEE Joint, 2002.

M. Sumner, *Enterprise resource planning*, 1st ed. Upper Saddle River, N,J.: Prentice Hall, 2005.

Pruftechnic Ltd, "Can Condition Monitoring Improve Your Plant Performance?," in *Worldpump*, 2004, pp. 40-47.

P.-S. Yeh and W. H. Miller, "A real time lossless data compression technology for remote-sensing and other applications," presented at Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on, 1995.

R. Itschner, C. Pommerell, and M. Rutishauer, "GLASS: Remote monitoring of embedded systems in power engineering," *IEEE Internet Computing*, vol. 2, pp. 46-48, 1998.

S. N. Laboratories, "History of SCADA," vol. 2006. [Online]. Available: http://www.sandia.gov/scada/history.htm, 2004.

W. L. Chan, A. T. P. So, and K. C. Li, "Real time data compression for power system monitoring using programmable gate array," presented at TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on, 1993.