A

Seminar report

on

“SCADA”

Submitted in partial fulfillment of the requirement for the award of degree

Of Bachelor of Technology in Computer Science

SUBMITTED TO:

www.studymafia.org

SUBMITTED BY:

www.studymafia.org

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly helped me in the seminar on “SCADA”.

I would like to express my gratefulness to……………… **,** who has given me the opportunity to carry out this seminar.

Lastly, I would like to thank the almighty and my parents for their moral support.

Abstract

SCADA stands for Supervisory Control and Data Acquisition. As the name indicates, it is not a full control system, but rather focuses on the supervisory level. It is a computer system for gathering and analyzing real time data. SCADA systems are used to monitor and control a plant or equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation. A SCADA system gathers information, such as where a leak on a pipeline has occurred, transfers the information back to a central site, alerting the home station that the leak has occurred, carrying out necessary analysis and control, such as determining if the leak is critical, and displaying the information in a logical and organized fashion. SCADA systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or incredibly complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system.

This paper describes the SCADA systems in terms of their architecture, their interface to the process hardware, the functionality and the application development facilities they provide.

Introduction

What is SCADA?

SCADA stands for Supervisory Control And Data Acquisition. As the name indicates, it is not a full control system, but rather focuses on the supervisory level. It is a software package that is positioned on top of hardware to which it is interfaced, in general via Programmable Logic Controllers (PLCs), or other commercial hardware modules. Systems similar to SCADA systems are routinely seen in factories, treatment plants etc. These are often referred to as Distributed Control Systems (DCS). They have similar functions to SCADA systems, but the field data gathering or control units are usually located within a more confined area. Communications may be via a local area network (LAN), and will normally be reliable and high speed. Basically, SCADA is a computer system for gathering and analyzing real time data.

What is data acquisition?

Data acquisition is the process of retrieving control information from the equipment which is out of order or may lead to some problem or when decisions are need to be taken according to the situation in the equipment. So this acquisition is done by continuous monitoring of the equipment to which it is employed. The data accessed are then forwarded onto a telemetry system ready for transfer to the different sites. They can be analog and digital information gathered by sensors, such as flow meter, ammeter, etc. It can also be data to control equipment such as actuators, relays, valves, motors, etc.

So why or where would you use SCADA?

SCADA can be used to monitor and control plant or equipment. The control may be automatic, or initiated by operator commands. The data acquisition is accomplished firstly by the RTU's (remote Terminal Units) scanning the field inputs connected to the RTU (RTU's may also be called a PLC - programmable logic controller). This is usually at a fast rate. The central host will scan the RTU's (usually at a slower rate.) The data is processed to detect alarm conditions, and if an alarm is present, it will be displayed on special alarm lists. Data can be of three main types. Analogue data (i.e. real numbers) will be trended (i.e. placed in graphs). Digital data (on/off) may have alarms attached to one state or the other. Pulse data (e.g. counting revolutions of a meter) is normally accumulated or counted.

These systems are used not only in industrial processes. For example, Manufacturing, steel making, power generation both in conventional, nuclear and its distribution, chemistry, but also in some experimental facilities such as laboratories research, testing and evaluation centers, nuclear fusion. The size of such plants can range from as few as 10 to several 10 thousands input/output (I/O) channels. However, SCADA systems evolve rapidly and are now penetrating the market of plants with a number of I/O channels of several 100K.

The primary interface to the operator is a graphical display (mimic) usually via a PC Screen which shows a representation of the plant or equipment in graphical form. Live data is shown as graphical shapes (foreground) over a static background. As the data changes in the field, the foreground is updated. E.g. a valve may be shown as open or closed. Analog data can be shown either as a number, or graphically. The system may have many such displays, and the operator can select from the relevant ones at any time.

SCADA systems were first used in the 1960s.SCADA systems have made substantial progress over the recent years in terms of functionality, scalability, performance and openness such that they are an alternative to in house development even for very demanding and complex control systems as those of physics experiments. SCADA systems used to run on DOS, VMS and UNIX; in recent years all SCADA vendors have moved to NT and some also to Linux.

Architecture:

In this section we are going to details which describe the common architecture required for the SCADA products.

Hardware Architecture

The basic hardware of the SCADA system is distinguished into two basic layers: the "client layer" which caters for the man machine interaction and the "data server layer" which handles most of the process data control activities. The data servers communicate with devices in the field through process controllers. Process controllers, e.g. PLC's, are connected to the data servers either directly or via networks or fieldbuses that are proprietary (e.g. Siemens H1), or non-proprietary (e.g. Profibus). Data servers are connected to each other and to client stations via an Ethernet LAN. Fig.1. shows typical hardware architecture.

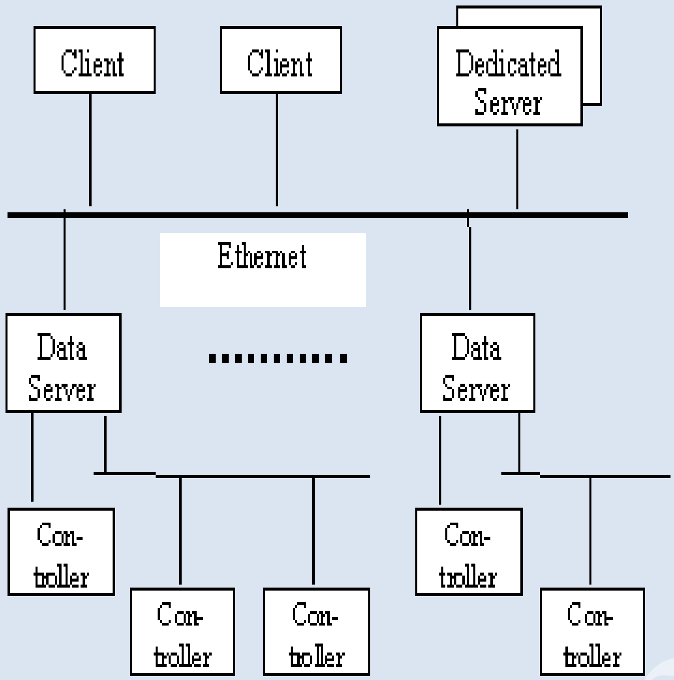

Figure 1: Typical Hardware Architecture

Software Architecture

The SCADA products are multi-tasking and are based upon a real-time database (RTDB) located in one or more servers. Servers are responsible for data acquisition and handling like polling controllers, alarm checking, calculations, logging and archiving) on a set of parameters, typically to which those are connected.

However, it is possible to have dedicated servers for particular tasks, e.g. historian, datalogger, alarm handler. Fig. 2 shows a SCADA architecture that is generic for the product.

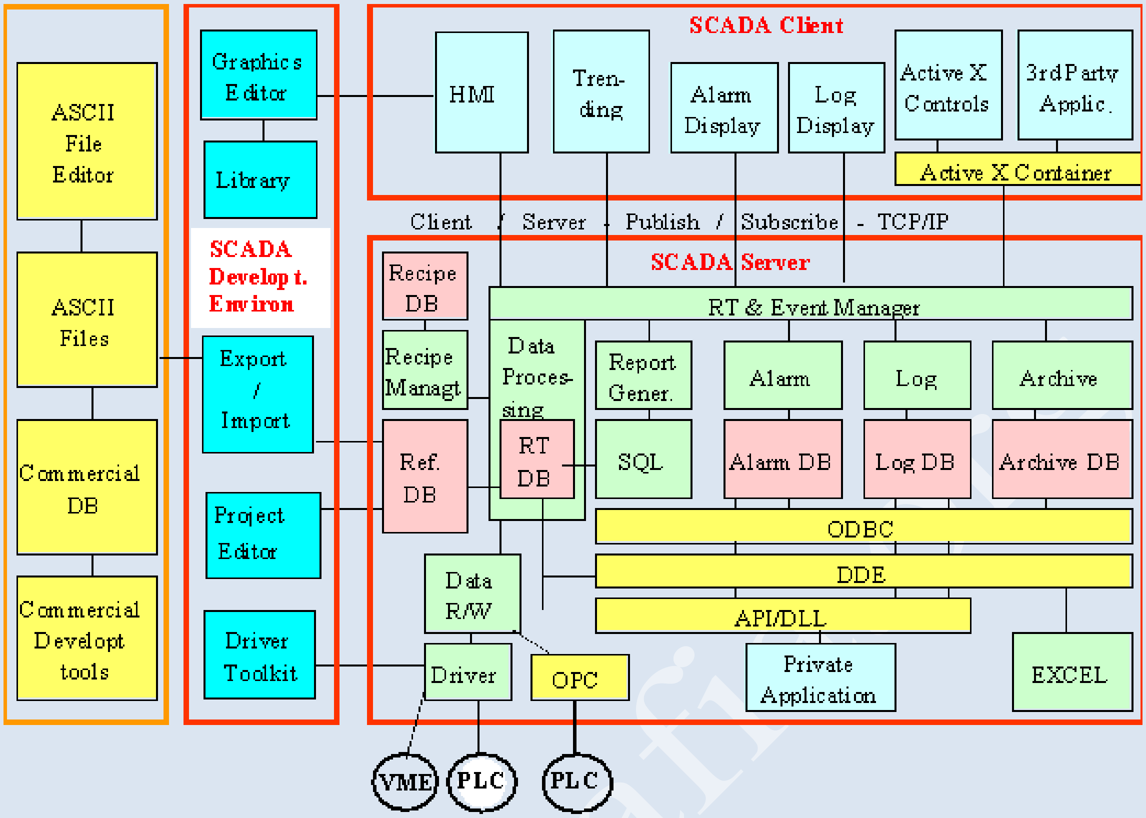

Figure 2: Generic Software Architecture

Communication:

Internal Communication:

Server-client and server-server communication is in general on a publish-subscribe and event-driven basis and uses a TCP/IP protocol, i.e., a client application subscribes to a parameter which is owned by a particular server application and only changes to that parameter are then communicated to the client application.

Access to Devices:

The data servers poll the controllers at a user defined polling rate. The polling rate may be different for different parameters. The controllers pass the requested parameters to the data servers. Time stamping of the process parameters is typically performed in the controllers and this time-stamp is taken over by the data server. If the controller and communication protocol used support unsolicited data transfer then the products will support this too.

The products provide communication drivers for most of the common PLCs and widely used field-buses, e.g., Modbus. Of the three fieldbuses that are recommended are, both Profibus and Worldfip are supported but CANbus often not. Some of the drivers are based on third party products (e.g., Applicom cards) and therefore have additional cost associated with them. VME on the other hand is generally not supported.

A single data server can support multiple communications protocols; it can generally support as many such protocols as it has slots for interface cards. The effort required to develop new drivers is typically in the range of 2-6 weeks depending on the complexity and similarity with existing drivers, and a driver development toolkit is provided for this.

Interfacing

Application Interfaces / Openness

The provision of OPC client functionality for SCADA to access devices in an open and standard manner is developing. There still seems to be a lack of devices/controllers, which provide OPC server software, but this improves rapidly as most of the producers of controllers are actively involved in the development of this standard.

The products also provide

- an Open Data Base Connectivity (ODBC) interface to the data in the archive/logs, but not to the configuration database,
- an ASCII import/export facility for configuration data,

- a library of APIs supporting C, C++, and Visual Basic (VB) to access data in the RTDB, logs and archive. The API often does not provide access to the product's internal features such as alarm handling, reporting, trending, etc.

The PC products provide support for the Microsoft standards such as Dynamic Data Exchange (DDE) which allows e.g. to visualize data dynamically in an EXCEL spreadsheet, Dynamic Link Library (DLL) and Object Linking and Embedding (OLE).

Database

The configuration data are stored in a database that is logically centralized but physically distributed and that is generally of a proprietary format. For performance reasons, the RTDB resides in the memory of the servers and is also of proprietary format. The archive and logging format is usually also proprietary for performance reasons, but some products do support logging to a Relational Data Base Management System (RDBMS) at a slower rate either directly or via an ODBC interface.

Scalability

Scalability is understood as the possibility to extend the SCADA based control system by adding more process variables, more specialized servers (e.g. for alarm handling) or more clients. The products achieve scalability by having multiple data servers connected to multiple controllers. Each data server has its own configuration database and RTDB and is responsible for the handling of a sub-set of the process variables (acquisition, alarm handling, archiving).

Functionality:

Access Control

Users are allocated to groups, which have defined read/write access privileges to the process parameters in the system and often also to specific product functionality.

MMI

The products support multiple screens, which can contain combinations of synoptic diagrams and text. They also support the concept of a "generic" graphical object with links to process variables. These objects can be "dragged and dropped" from a library and included into a synoptic diagram. Most of the SCADA products that were evaluated decompose the process in "atomic" parameters (e.g. a power supply current, its maximum value, its on/off status, etc.) to which a Tag-name is associated. The Tag-names used to link graphical objects to devices can be edited as required. The products include a library of standard graphical symbols, many of which would however not be applicable to the type of applications encountered in the experimental physics community. Standard windows editing facilities are provided: zooming, re-sizing, scrolling... On-line configuration and customization of the MMI is possible for users with the appropriate privileges. Links can be created between display pages to navigate from one view to another.

Trending

The products all provide trending facilities and one can summarize the common capabilities as follows:

- the parameters to be trended in a specific chart can be predefined or defined on-line
- a chart may contain more than 8 trended parameters or pens and an unlimited number of charts can be displayed (restricted only by the readability)
- real-time and historical trending are possible, although generally not in the same chart
- historical trending is possible for any archived parameter
- zooming and scrolling functions are provided
- parameter values at the cursor position can be displayed

The trending feature is either provided as a separate module or as a graphical object (ActiveX), which can then be embedded into a synoptic display. XY and other statistical analysis plots are generally not provided.

Alarm Handling

Alarm handling is based on limit and status checking and performed in the data servers. More complicated expressions (using arithmetic or logical expressions) can be developed by creating derived parameters on which status or limit checking is then performed. The alarms are logically handled centrally, i.e., the information only exists in one place and all users see the same status (e.g., the acknowledgement), and multiple alarm priority levels (in general many more than 3 such levels) are supported.

It is generally possible to group alarms and to handle these as an entity (typically filtering on group or acknowledgement of all alarms in a group). Furthermore, it is possible to suppress alarms either individually or as a complete group. The filtering of alarms seen on the alarm page or when viewing the alarm log is also possible at least on priority, time and group. However, relationships between alarms cannot generally be defined in a straightforward manner. E-mails can be generated or predefined actions automatically executed in response to alarm conditions.

Logging/Archiving

The terms logging and archiving are often used to describe the same facility. However, logging can be thought of as medium-term storage of data on disk, whereas archiving is long-term storage of data either on disk or on another permanent storage medium. Logging is typically performed on a cyclic basis, i.e., once a certain file size, time period or number of points is reached the data is overwritten. Logging of data can be performed at a set frequency, or only initiated if the value changes or when a specific predefined event occurs. Logged data can be transferred to an archive once the log is full. The logged data is time-stamped and can be filtered when viewed by a user. The logging of user actions is in general performed together with either a user ID or station ID. There is often also a VCR facility to play back archived data.

Report Generation

One can produce reports using SQL type queries to the archive, RTDB or logs. Although it is sometimes possible to embed EXCEL charts in the report, a "cut and paste" capability is in general not provided. Facilities exist to be able to automatically generate, print and archive reports.

Automation

The majority of the products allow actions to be automatically triggered by events. A scripting language provided by the SCADA products allows these actions to be defined. In general, one can load a particular display, send an Email, run a user defined application or script and write to the RTDB.

The concept of recipes is supported, whereby a particular system configuration can be saved to a file and then re-loaded at a later date. Sequencing is also supported whereby, as the name indicates, it is possible to execute a more complex sequence of actions on one or more devices. Sequences may also react to external events. Some of the products do support an expert system but none has the concept of a Finite State Machine (FSM).

Evolution:

SCADA vendors release one major version and one to two additional minor versions once per year. These products evolve thus very rapidly so as to take advantage of new market opportunities, to meet new requirements of their customers and to take advantage of new technologies.

As was already mentioned, most of the SCADA products that were evaluated decompose the process in "atomic" parameters to which a Tag-name is associated. This is impractical in the case of very large processes when very large sets of Tags need to be configured. As the industrial applications are increasing in size, new SCADA versions are now being designed to handle devices and even entire systems as full entities (classes) that encapsulate all their specific attributes and functionality. In addition, they will also support multi-team development.

As far as new technologies are concerned, the SCADA products are now adopting:

- Web technology, ActiveX, Java, etc.
- OPC as a means for communicating internally between the client and server modules. It should thus be possible to connect OPC compliant third party modules to that SCADA product.

Potential benefits of SCADA:

The benefits one can expect from adopting a SCADA system for the control of experimental physics facilities can be summarized as follows:

- A rich functionality and extensive development facilities. The amount of effort invested in SCADA product amounts to 50 to 100 p-years!
- The amount of specific development that needs to be performed by the end-user is limited, especially with suitable engineering.
- Reliability and robustness. These systems are used for mission critical industrial processes where reliability and performance are paramount. In addition, specific development is performed within a well-established framework that enhances reliability and robustness.
- Technical support and maintenance by the vendor.

For large collaborations, using a SCADA system for their controls ensures a common framework not only for the development of the specific applications but also for operating the detectors. Operators experience the same "look and feel" whatever part of the experiment they control. However, this aspect also depends to a significant extent on proper engineering.

References:

www.ref.web.cern.ch/ref/CERN/CNL/2002/003/scada/

www.studymafia.org

www.scadanews.com

Wikipedia

www.scada.com