BUILDING A REDUNDANT DATA ACQUISITION SYSTEM USING IDE AND PLC SIMULATION

[1]LEENA M. SANAP, [2]SWATI CHAUDHARI, [3]BHARTI SINGH

[1,2,3]K.J. Somaiya College of Engineering, Mumbai, India, P.E.S. Modern College of Engineering, Pune, K.J. Somaiya College of Engineering, Mumbai, India.
E-mail: [1]leena.sanap@somaiya.edu, [2]swatichaudhari02@gmail.com, [3]bhartisingh@somaiya.edu

Abstract— This paper refers to the how a redundant system can be built for a process plant without hindering its function. It makes use of advanced SCADA IDE (Integrated development environment) and PLC simulation software in order to test the successful implementation. It involves various aspects for a secure process in a plant like alarms, trends and historizes data.

Index Terms— Alarms, historize, IDE, PLC simulation, redundant system, Trends.

I. INTRODUCTION

Automation has become a vital part of industry today. It equips producers for mass production with better quality. Automation systems eliminate the need of human assistance by involvement of feedback and sensory programs. Sridhar Raja in his paper on building automated systems has mentioned the distinct importance of PLC and SCADA for automation, control and monitoring [2]. A paper on smart car washing system has discussed the benefits of PLC, like running multiple machines, troubleshooting, no rewiring and visual operation on screen [3]. A paper on geothermal plant depicts the vital part played by PLC and SCADA [6]. This in turn brings to light as to how much important it is for the system to be accurate and efficient. There should be a minimal or no possibility that the system may fail and also if it does at all it should begin to function in a minimum time lapse. For ensuring this various techniques and procedures are implemented everywhere. One of them is redundancy. A substantial amount of literature is available giving information about the various types of redundancies that can be implemented. While there are various methods, techniques, and terminologies for implementing redundancy, the following models represent the more common ones used in industry. The three main models are Standby Redundancy, N Modular Redundancy, and 1: N Redundancy [4].
The following is a general model showing how data is transferred from operator end to the plant.

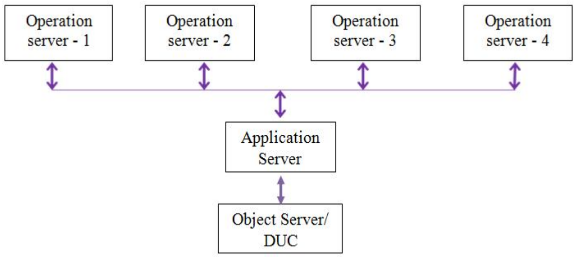

Fig.1 An ideal non-redundant model

This model depicts a three tier system which is prone to a lot of run-time failures. The entire load is handled by a single application server that makes the system risky.

II. REDUNDANCY

Redundancy in a network is the additional or alternate instances of network devices, equipments or means of communication that are installed in a framework in order to ensure availability in case of a path failure. It is a backup mechanism for uninterrupted functioning of a system.
When it comes to engineering control systems, there are various models of redundancies available. Use of each of it depends on the priority of the process and how critical its function is. The various divisions and sub-divisions of redundancy techniques are as follows:

1. Standby redundancy

This type of redundancy makes use of a backup or spare network/device hence the name. The secondary unit developed is generally a duplicate of the primary which can takeover temporarily or for the period of replacement or error correction is carried out. The most important concern in this technique is the synchronization of the two units also the bump or glitch that may occur during the switching.

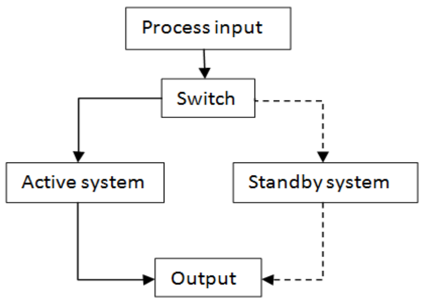

Fig.2 Standby Redundancy

It is further divided as:

A. Cold Standby

In this technique, whenever a system fails i.e. the active system is unable to control the ongoing process the primary unit is shut down. The secondary unit that is intended to take over in order to smoothly function the plant is to be powered ON. The drawback of this design is that the downtime is greater than in hot standby, because you have to power up the standby unit and bring it online into a known state. This makes it more challenging to reconcile synchronization issues [4]. It can be said that cold standby can be used for process which are less time critical and allow human intervention.

B. Hot Standby

In case of hot standby, the secondary unit also receives the data simultaneously as the primary but it does not take part in analyzing or delivering the data to output. Whenever the active system fails, the secondary takes over immediately and since it is in synchronization, the process is uninterrupted. Optionally the standby unit can be made to monitor the failed unit so as to switch back when the fault is corrected. If active system failure, spare system will take over control process from active system automatically hut the outputs of controllers need time to recover state. If recover time is more than permission downtime then process must be shutdown [1].

Considering all the above cases it is helpful to make use of hot standby redundancy for this project. The data transfer is very crucial from the mid level of the model to the third level thus the features of hot standby that is immediate switching supports the process. The only challenge is to synchronize both the primary and standby servers.

III. PROPOSED MODEL

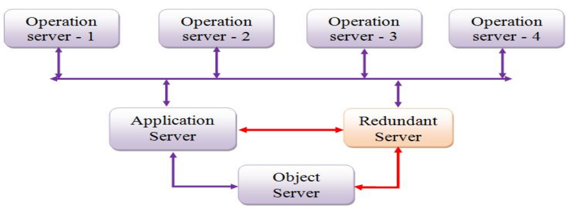

Fig. 3 Proposed model.

This system is a real-time implementation.

The application server is like the heart of the system. It is responsible for all the data entering/leaving the system, the operating functions and processes. The Redundant Application Server should be such that there can be synchronized data maintenance with the functioning application server. The implementation of redundancy would be at two points:

1. In between the operating server and the object server
2. In between the Application server and the object server/ PLC.

Customized and live *alarms, historical and real time trends*_can be implemented. The application server handles all the data. Thus if this server is lost, client loses connection to all the data. The solution to this is installing another server which maintains synchronized data through a dedicated network. Next point of redundancy arises if there is a loss of control by the application server. This would cause the server to lose connection with the field/object server. Solution for this is installing a standby network which takes over as soon as active network loses control. Whenever a failure occurs in the production line or field, a heavy loss has to be incurred. The process to detect the point of failure is time consuming for larger plants. Also if the means of operation involves parts at different locations, it is difficult. In this case it becomes necessary to develop a mechanism that would issue an alarm as soon as any unintended event occurs in production. Also the point of failure i.e. the cause should be detectable. Also the possible actions that the operator can take to avoid the accident should be advised.

IV. METHODOLOGY

For implementing the above said model, we make use of Archestra IDE (Integrated development environment) which is an advanced SCADA, InTouch which is HMI (human-machine interface) software and in place of PLC, InControl version 7.11 is used.

A) InControl 7.11

InControl is a real-time application engine which runs a wide range of manufacturing processes that require high speed, deterministic processing of data and logic. It provides high functionality in the sense that it supports all type of PLC programming languages like Relay Ladder Logic (RLL), Sequential Function Chart (SFC) and Structured Text (ST) editors. For this project it is necessary to check whether the data reaches uninterruptedly even when the switching has occurred. Thus we implement the ladder logic in InControl and interface it with Archestra.

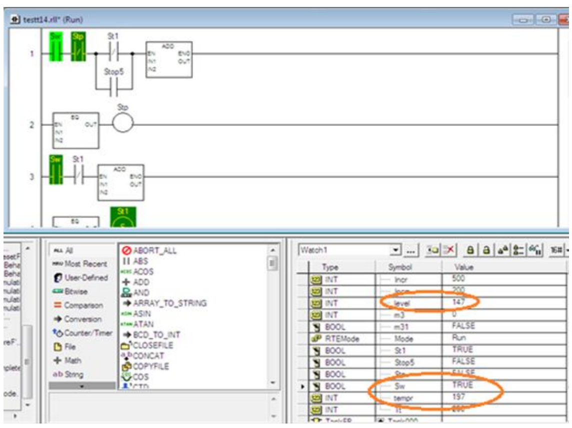

Fig 4. Ladder logic implemented in In Control.

As seen above we can know the current value of a particular attribute of interest say level, temperature etc. Thus it becomes easier for monitoring.

B) About InTouch

We make use of Wonderware InTouch 2014 R2 (version 11.1) since it supports modern applications i.e. native support of Archestra IDE besides wizards. InTouch is an HMI software which has inbuilt tools which makes it very easy to customize every component in the plant in an industry.It works on the assignment of tags. i.e each of the attribute or variable is represented in the form of a tag. The following are the three main components of InTouch:

1. Application manager
2. Window Maker
3. Window Viewer

The application manager organizes the application created. Information about the application is displayed in columns that include name, path, resolution, version, mode and description. Operations like DBDump, DBLoad, import, export can be performed here.

Window Maker is the main development area used to create InTouch applications where object oriented graphics are used to create animated, touch-sensitive display windows. It also involves writing Quickscripts and customizing properties of the objects used. It facilitates smart symbol creation wherein it is possible to create a particular object as per requirement by making use of tools like paint. The assignment of tags is carried out in the window maker. It is very important to select the correct tag type when assigning. To name a few the tag types are of the type memory and i/o where i/o tags are used when data is to be received through an interface or from an external source. Memory tags are used for acquiring and processing data internally. Also it is possible to give animation links like blinking, orientation, percent fill, hide/show window and even write action script for a particular object or a part of that object.

Window Viewer is the runtime environment used to display the graphic windows created in the Window Maker. It executes scripts, performs historical and alarm data logging and reporting. It can be said to be the operator end since this is what will be visible to the operator when interacting with the system.

C) About Archestra IDE

Archestra is a system platform which is divided into parts like the historian server, information server and the InTouch View App. Due to this the load is shared as the task is now distributed and it gets integrated on a single platform providing higher accuracy and improved efficiency. Working with Archestra begins by creating a galaxy. The galaxy gets stored at the specified location whose path is to be provided. Archestra works on the basis of a hierarchical structure which is as follows:

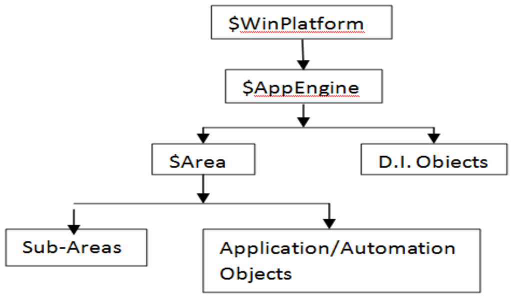

Fig 5. The Archestra hierarchy

$WinPlatform, as the name suggests is at the top of this hierarchy. It is a representation of a computer in the automation application. It is necessary to specify the node that is the i.p address of the server from which data is obtained if the computer which provides the data is the one where galaxy is created then localhost must be entered. $AppEngine hosts and schedules execution of application objects, areas and device integration objects. It has in-built templates which help create analog/ discrete devices.

$Area organizes all other types of automation objects in a logical manner that mimics the actual physical facility. It represents a plant area and allows grouping of objects for modeling and alarm reporting. It allows sub-nesting upto 10 levels.D.I. object means the device integration object. The redundancy of the servers can be implemented by first creating a $RedundantDIobject which has to be configured such that it decides which of the sub-objects should be treated as active and standby. This template will host the two server configurations which can be created by making use of $DDESuitelinkClientobject. The most important part of configuration is the i.p address. It should match with the computer where InControl is running. The server name should be specified as RTEngine. Both these active and standby require specification of an item list which contains a list of the tagnames and corresponding variable names assigned while addressing of ladder logic.

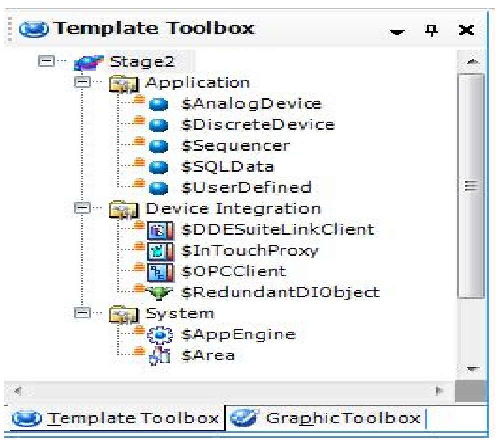

Fig. 6 Parent templates utilized for building a system

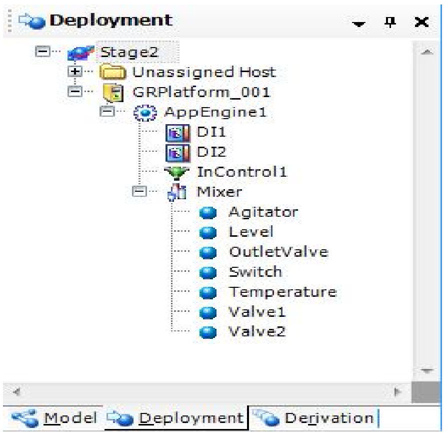

Fig. 7 Deployment view of the implemented system.

$InTouchViewApp template represents a managed InTouch application which means we can perform every function that exists in the InTouch software through Archestra. Also all the graphics that are created are assembled under these by making use of the graphic toolbox.In paper for smart traffic control system, they have made use of the InTouch software to show congestion and diversion in city as well as traffic status and diversion for vehicles at toll booth [5].

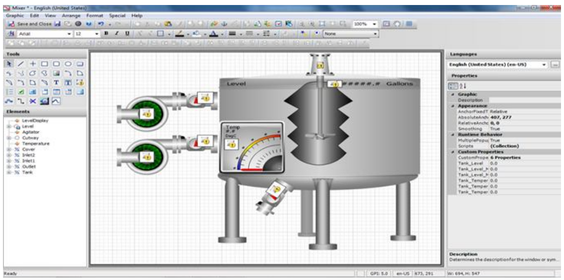

Fig. 8 Graphics showing tank with cutway, valves, agitator, pumps designed in InTouch.

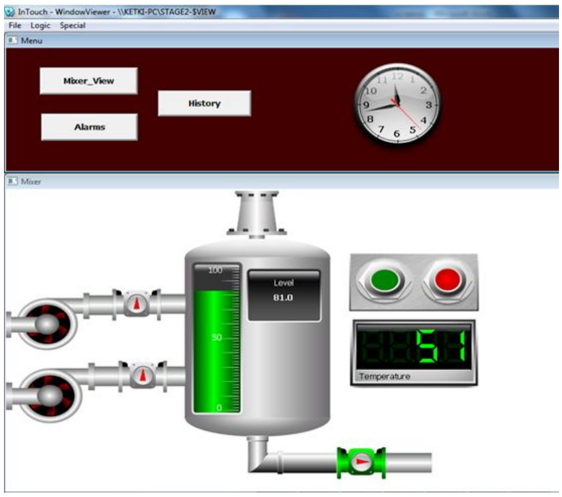

Fig. 9 Graphics showing the current position of the system along with level, temperature parameters.

In the above figure we observe the on and off switch which are configure through scripting and made by making use of different objects like LEDs' back plate etc.

Fig. 10 Showing the active server.

Fig. 11 Connection status of server on failover.

It is possible to include the timing and cause of failure of the server in the graphics. Also the number of times the switch occurrences between active and standby can be recorded.

V. SOME OTHER FEATURES

1. Alarms

Alarms represent warnings of process conditions that could cause problems and require an operator response. A typical alarm is triggered when a process value exceeds a user-defined limit, such as an analog value exceeding an upper threshold. This triggers an alarm to notify the operator of a problem. Within the InTouch HMI, alarms are classified into general categories based on their characteristics. They are known as Class and Type. The Distributed Alarm system categorizes all alarms into following general conditions:

A) Discrete
- (ON/OFF)

B) Analog
- Value (LoLo, Lo, HiHi, Hi)
- Deviation (Major, Minor)
- Rate-of-Change. (%mD, %MD)

Among other additional features configuration is possible for alarm inhibitors, generating alarm query, alarm displays and alarm database management system. An alarm can be inhibited in InTouch by assigning to each alarm or alarm sub-state an inhibitor alarm tag that does not allow the alarm from transitioning into an active state. We can generate an alarm query which filters only those alarms that the user or even developers wishes the operator to view. The particular file thus generated can also be saved in a ".xml" file format. The Distributed Alarm Display object provides a single display to show both local and remote alarms. This display object's features include built in scroll bars, sizable columns, multiple selection of alarms, a status bar, a shortcut menu, and colors based on alarm priority and state. The SQL Access Manager can be used to transfer data, such as batch recipes from a SQL database to an InTouch application. SQL Access Manager can also be used to transfer run-time data, alarm status, or historical data from InTouch to a database. The application tool Alarm DBLogger The file thus created has the extension ".LGH".

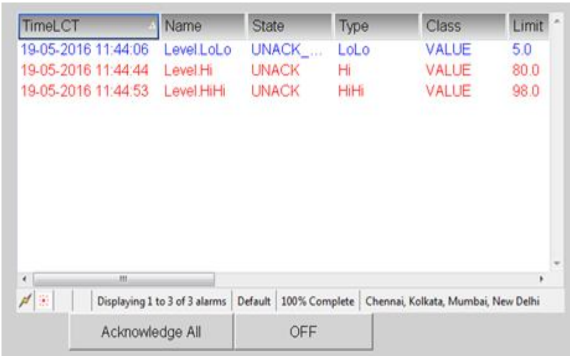

Fig. 12 Alarm window showing alarms with date and time of occurrences.

2. Trends

Historian server is used for obtaining the trend of a particular parameter. It is necessary to enable PV of that object and historian in the WinPlatform of the system. From the tagpicker we have to add the particular attribute whose trend is to be observed.

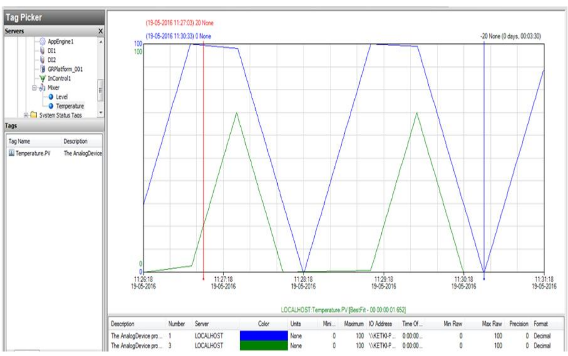

Fig. 13 Trend generated for parameters of level and temperature.

CONCLUSION

This project attempts implementation of an efficient redundant system for acquiring data in real-time for an industrial plant. The proposed model architecture is simple yet effective in achieving the desired goal of redundancy. Only by making changes in the network and allowing data to be reachable at multiple nodes secure system is set up as compared to the existing system. The alarms that appear should be of high accuracy and details about corrective action taken should also be notified to the operator. Trends keep a track of the data flow. Security at different levels controls the access to and from the plant. For upgrading a process in plant, simulation can be performed in InControl before loading it in PLC. Information about the active/standby server and cause of failure of one, along with the timing details plays an important role for accurate functioning of the plant. The whole system becomes highly effective with the aid of graphics. The more it is visual the more is the convenience. Thus after implementation of this system the fault tolerance will increase substantially.

REFERENCES

[1] V. Tipsuwanpom, A. Sangrayub, T. Suesut, A. Numsomran, S. Gulphanich, "Development of PLC Fiber-optic Network for Redundant System", IEEE ICIT'02, Bangkok, Thailand.

[2] Sridhar Raja .D, Suresh Kumar, "PLC application in automated Building Management System", IJAREEIE Vol. 3, Issue 4, April 2014.

[3] Akilandeswari.K , Haripriya.J ,Sangavi Nirmala.V ,Rathna Prabha.S , "Implementation of Smart Car Washing Using GSM PLC", IJAREEIE Vol. 3, Issue 3, March 2014.

[4] "Redundant System Basic Concepts", White paper, National Instruments, Publish Date: Jan 11, 2008

[5] Mohit Dev Srivastava, Prerna, Shubhendu Sachin, Sumedha Sharma, Utkarsh Tyagi, "Smart traffic control system using PLC and SCADA ", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 1, Issue 2, December 2012.

[6] Dr. S. Senthil Kumar, Associate Professor, Dept. of E&I, Bharath University, Chennai, "Geothermal Power Plant Design using PLC and SCADA", ISSN 2278 – 8875, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.Vol. 1, Issue 1, July 2012.

★★★