Computer Standards & Interfaces 31 (2009) 599–612

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Web based multilayered distributed SCADA/HMI system in refinery application

Adnan Salihbegovic [a,*], Vlatko Marinković [a], Zoran Cico [a], Elvedin Karavdić [b], Nina Delic [a]

[a] ETF Sarajevo, Bosnia and Herzegovina
[b] Bosna-S Oil Services Company, Bosnia and Herzegovina

ARTICLE INFO

Available online 9 April 2008

Keywords:
SCADA/HMI
Data server
Fieldbuses
OPC protocols
Real time systems

ABSTRACT

The paper describes system synthesis and architecture of a multilayered distributed SCADA/HMI system. The system is used for monitoring and control of refinery terminals for truck loading and oil products pipeline shipping. Network-centered, distributed PLC system with SCADA functions and several levels of fieldbuses, interconnected with the HMI part of the system is described first. Following is a brief description of the software tools used in the system design. Configuration and PLC controllers programming, data server configuration, operator panels design and way of managing the control system using a standard Web browser, e.g. Internet Explorer, are addressed. After examining the way specific customization is done, the paper focuses on establishing the link with icons on graphic panels that integrates all three levels of process control from Operator's panels.

1. Description of the basic system

A distributed PLC based SCADA/HMI system was built with the standard elements of Yokogawa Stardom system—a networked-based Control System (NCS) that is by its architecture and functional capabilities positioned between the lower-end PLC/SCADA systems and the higher-end DCS [1].

This means that it includes and is capable of doing everything PLC based SCADA systems can do, having powerful FCN/FCJ PLC controllers and Java based HMI software VDS (Versatile Data Server). Due to its advanced networking concepts and modular structure, the STARDOM system can be expanded functionally and geographically to build full fledged DCS systems.

The following Fig. 1 illustrates the architecture of the STARDOM system:

From the figure above, one can see that it is a modular component-based system, in which each component can be used independently of the others. However, it can also be fully integrated so that all components programmatically connect among themselves. Furthermore, the tag database can be globally defined, with more powerful PLC/SCADA and DCS systems [2].

The main features of the system that were used in the application design and synthesis, are:

Control functions

- *IEC61131* standard for programming PLC controllers, which enables entering of the control structures in one of the five programming languages:
  - Function Blocks (FB)
  - Ladder Diagrams (LD)
  - Sequence Functions Charts (SFC)
  - Structured Text (ST)
  - Instruction List (IL)
- *Application Portfolios*—user-defined programs that can be security protected for use in other projects. It is based on the use of a layering structure so that function blocks can be created from other function blocks and be in turn used in other function blocks as user-defined library modules. Application Portfolios can be created from a set of function blocks to define a simple or complex control strategy, and can be re-used in other projects, implementing the idea of reusable software modules.
- *Embedded HMI Server* and email function in the field controllers, providing autonomous functionality of the controller. No SCADA HMI (such as the VDS) is required, making them suitable for embedded and remote applications.
- *Full redundancy* in controller hardware. Power supply, CPU and network connections can all be made redundant for increased system availability [3].

* Corresponding author.
E-mail addresses: adnan.salihbegovic@etf.unsa.ba (A. Salihbegovic), vmarinkovic@gmail.com (V. Marinković), zorancico@yahoo.com.au (Z. Cico), elvedin.karavdic@bosna-s.ba (E. Karavdić), ninadelic@gmail.com (N. Delic).

0920-5489/$ – see front matter
doi:10.1016/j.csi.2008.03.023

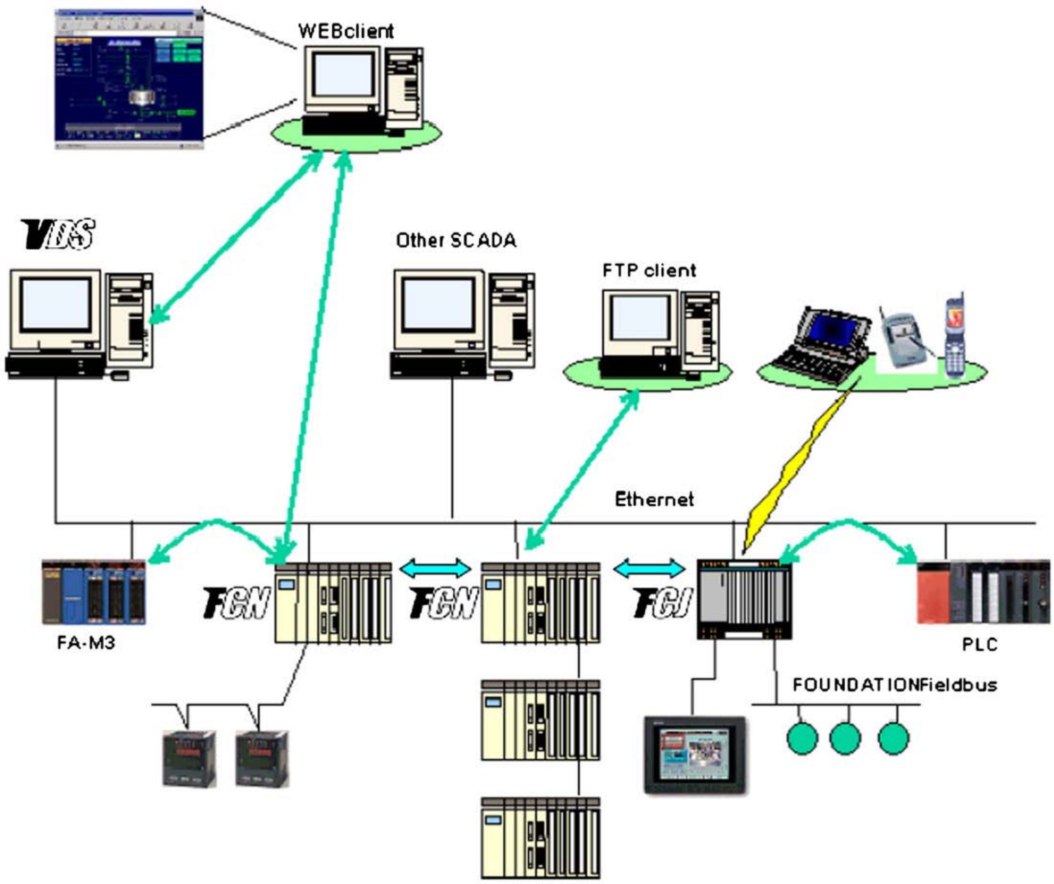

Fig. 1. Topology of the Stardom system.

Network functions

- *HSE* (high speed Ethernet) with 10/100 Mbps, communications between controllers, up to five data servers and PC-based HMI stations.
- *OPC* communications between the PLCs and other devices via a PC-based OPC gateway. OPC Client and Server functions are available in the VDS which is in itself built on this communication protocol.
- *Standard Web interface* to operator displays, i.e., the operator stations are PCs with standard Web browser, and no proprietary software is required.
- *Field Networking*—the controllers communicate with field transmitters, other PLCs, and I/O devices over Foundation Fieldbus, Profibus, Modbus and a range of other field networks [4–6]. Generic communications to serial and Ethernet devices (e.g., barcode reader, RF-ID reader, etc.) can be readily implemented in the control program [7,8].

Data server functions

The VDS can communicate to third party devices, as well as to the PLCs referred to as controllers. This provides integration of the control hardware on site to a central monitoring and control point. This means that a DCS type global database, which ensures easier tag management, can be implemented [9].

1.1. PLC controller

This is a rack-mounted unit with up to three racks of I/O modules. I/O modules also include communication modules for remote I/O, PLCs of other manufacturers, Modbus and Fieldbus. It is programmable with IEC61131 standard programming languages including Java, and is built on the Intel 80186 processor. The software is organized around VxWorks 6.0, a real time operating system, which guaranties effective, and reliable application programs control and management even for hard real time tasks with fixed deadlines [10]. During the described application development, various schemes for tasks configuration and memory allocation policies based on best fit algorithm were tested [11]. It was done in order to tune for optimal system response and to achieve fastest execution time of periodic tasks, such as data acquisition of measurement values and feedback control loop updates [12]. Since within the applications included in the Project, there were many safety critical functions, the correctness of selection of VxWorks as real time operating system was tested in many safety related situations specified in hazop and worst case scenario that were carefully defined in the Project [13,14]. Fig. 2 below shows this PLC controller in a double redundant configuration with the range of I/O modules.

1.2. HMI software (VDS—Versatile Data Server)

This software suite is a Windows based application that combines the functions of a data server and a HMI Server to collect data from PLCs, Yokogawa and other manufacturers, I/O modules and DAQ Stations, and third party devices. It provides information about acquired data to the user via web based operator displays. The data server and HMI Server connect to each other using the OPC protocol, and can run on the same or separate PCs. Third party applications can also access the data server using OPC [1,15].

Fig. 2. Double redundant PLC controller.

1.3. Engineering tools

A suite of engineering tools available was used to program the field controllers and the VDS. These include:

- Logic Designer—which is a full IEC61131 complying programming interface
- Resource Configurator—hardware configuration tool
- VDS Builder (Object Builder, Graphic Designer)—consists of the set of programming tools for creating data server database, tools for operator screens design (graphic screens) using graphic editor and tools for overall system configuration, testing and debuging.

1.4. Project layout

Complete project (control system) code resides at three locations (Fig. 3):

- PLC Builder—created and edited by Logic Designer
- Data server Builder—created and edited by Object Builder and Graphic Builder
- Web server—created by HMI Deployment tool

1.5. Software structure of the system

The diagram on Fig. 4 below depicts the basic functions, applications and data flows inside the PLC controllers and the Versatile Data Server (HMI), including software tools for system configuration, control logic and graphic design.

2. Control system implementation for truck-loading terminal and oil products pipeline shipping

The refinery terminal, where the described system was implemented, consists of 6 main processing areas: Truck-loading Gantry, Pump Station A (6 loading pumps), Pump Station B (11 shipping pumps), Electrical Substation, Tank Farm and Administrative Building with Control Room. Different products, like fuel oil, gas oil, premium, regular, kerosene, avgas and LPG are coming from the refinery, and are used for tank stocking, truck-loading and pipeline shipping.

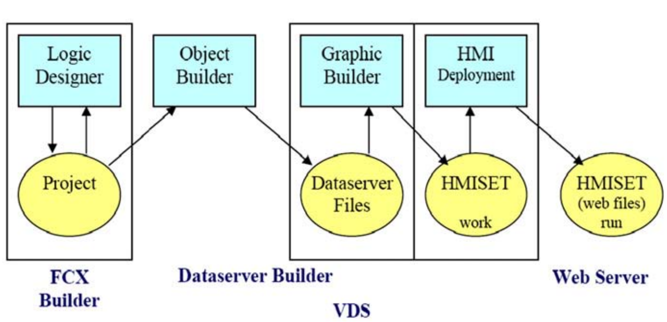

Fig. 3. Software architecture.

PLC based SCADA/HMI system, was used for control and monitoring of all processes on the terminal, enabling automatic loading sequences and generation of the reports of loaded products to trucks, measurement of all process variables, indication and material balances of all incoming and outgoing products, tank inventories, automatic control of process variables (temperatures, pressures, densities, etc), alarm registration, acknowledgement and archiving, real time trending and trend history windows and many more [16]. Controller CPUs, power supply units and communication ports are all selected to be dual redundant, providing additional reliability and availability to the system [1,3].

The truck-loading gantry consists of 11 loading arms, used to load gas oil, premium, regular, kerosene and heavy naphtha to trucks. The products are pumped to the loading gantry and trucks by six loading pumps in pump station A. Heavy naphtha is transferred to the loading gantry directly from the refinery. Upstream of each loading arm, a PD flow meter, flow computer and a set/stop valve are installed, providing automatic loading sequence and precise measurement of loaded quantity [3]. RS-485 serial communication is used to connect all 11 flow computers with PLCs, enabling automatic starting of the loading pumps when request for loading from loading gantry is received, and also enabling the complete monitoring of the loading system, including on-demand and scheduled reports. Pump station B consists of 11 shipping pumps, which are used to transfer products through 2

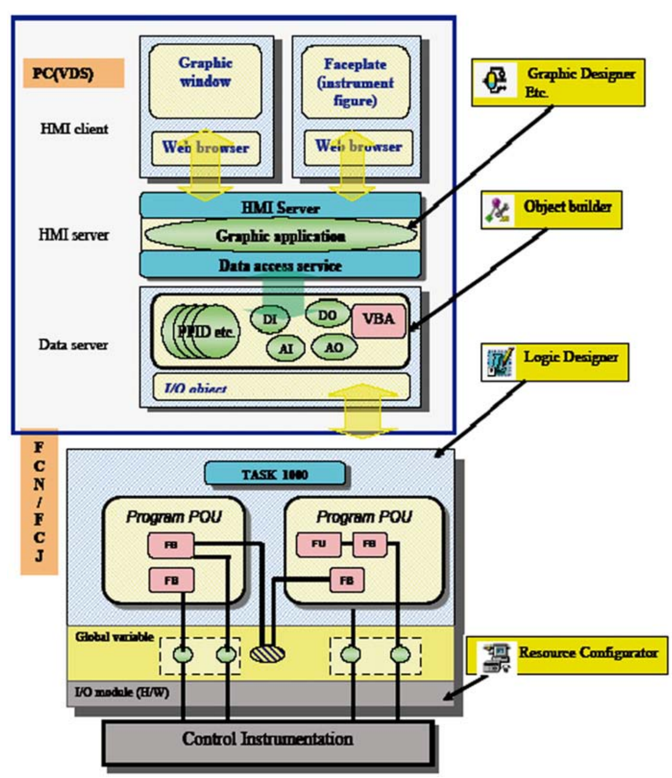

Fig. 4. Data flow through software layers.

pipelines. The white products pipeline is used to transfer gas oil, premium and LPG to the product-loading harbor for overseas tanker transportation, through a 40 km long pipeline.

The Fuel oil pipeline is used to transfer heated fuel oil and gas oil to an electricity power plant through a 30 km pipeline.

All incoming and outgoing quantities are custody metered by FMC/Smith turbine flow meters and PLCs enable indication, totalization and reporting of all flow rates and quantities. Furthermore, user-defined control blocks enable automatic and manual control of all MOVs (motor operated valves) and pumps, in loading and pumping sequences.

All MOVs at the terminal are interconnected in the Rotork Pakscan network, enabling control and monitoring of actuators using just one twisted pair wire. The network is closed in a ring, making it dual redundant and providing additional reliability [17,18].

The tank farm consists of several product tanks, for premium and gas oil storage, and two LPG spheres. Enraf radar level gauges are used to measure product levels [19], and a dedicated communication interface unit (CIU) is used to transfer the measurement data to the central SCADA/HMI system [20].

2.1. Control system design

Fig. 5 below depicts the topology of the implemented SCADA/HMI control system at the refinery terminal.

The control system is housed in two control cabinets, Cab1 in a control room and Cab2 in an electrical substation. There is one FCN controller (with dual-redundant CPU, network cards, power supplies and buses) in both of these cabinets and these controllers are interconnected using an Ethernet communication through a fibre-optic cable. The operator's Console inside the control room includes an industrial PC which runs VDS (e.g. Data Server and HMI Server).

There are also two Touch Screen Panel PCs mounted on the front side of the control cabinet Cab1. By using these panels, operators can monitor and control all system functions.

Furthermore, there is an engineering station based on the large size LCD monitor and industial PC which runs the VDS server software and is located at the console. Together with the maintenance station, there are in total four different operator screens in the control room that can be watched and used at the same time. All the other PC-based, PDA or notebook-based Operator stations are connected as thin clients either wirelessly in the field, or through a VPN network over the Internet [21].

2.2. Software configuration of the control system

Appplication software development and configuration is done based on system functional specification and underlying hardware configuration. Resource Configurator was used for hardware modules configuration and for the definition of every input and output signal from the field being directly connected to the system, and not using fieldbuses. Subsequently, by using Logic Designer, two projects have been created, FCN001 and FCN002, specifically for each hardware configuration in both cabinets. The earlier defined I/O configuration from Resource Configurator, has been copied to a Device Label Definition file inside Logic Designer with defined data ranges and engineering units for every analogue signal. After editing these data, all I/O tags were defined as global variables and were used during control logic development inside Logic Designer.

In order for these process tags to be available in Data Server, Logic Designer's tag database configuration, needs to be exported to Object Builder. This configuration is automaticaly saved to the ADLST.csv file after every compilation of the project instance in Logic Designer. In design project, two configuration files were saved in two different folders, named Cab1 and Cab2. After importing these files to Object Builder, the folder name is automatically used as a prefix for each defined control object. In this way, the origin of every tag is always known, which is convenient for further program development and system and software maintenance [22].

After most of the tag database is up and running inside Data Server, development of the graphic operator screens can start. Graphic Designer used for this Project is similar to other graphic tools for HMI graphic window design, although missing some important capabilities like multilayered design and vector graphics. Specific for this tool is Real Time Alarm window, which does not require any special configuration or settings. It automaticaly collects all the data defined inside Logic Designer and detects their violation of configured alarm limits, generating process alarms and showing their status to the operator. It is also possible to select sound output, which will notify Operators on occurrence of any alarm, perpetuating the sound signal until the operator acknowledges it.

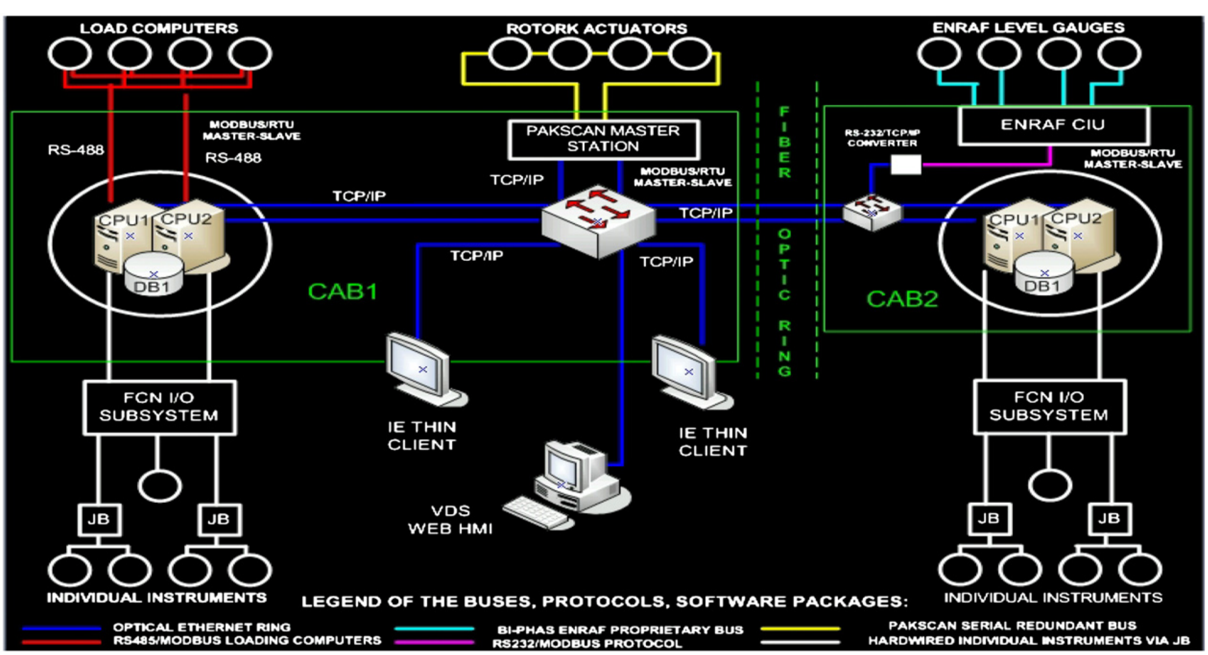

Fig. 5. Overall system topology.

Regarding the graphical options available, every label, indicator or any other element and its attributes can be linked with tags from Data Server using the DataSource Dialog option. Another function which we frequently used was Graphic Modify. With this option, graphic objects can change their properties according to the values of the process tags in Data Server.

2.3. Control logic and user-defined control blocks

User-defined control blocks, have been created as generic for control of pumps and MOVs to enable reusing the same block multiple times, thus creating the modular structure of the control logic system and speeding up design. In this way, the same logic block was used for the control of all 60 MOVs, the only difference being input and output tags connected to the block. The same user-defined control block was created to control all 17 pumps in the Project.

2.4. User-defined block for MOV control

The structure of the MOV control block is shown on Fig. 6.

The same generic block is used for all type of on/off MOVs, except for the 6 control MOVs (actuators coupled with control valves). Using this block and its associated faceplate, Operator can carry out automatic/manual and open/stop/close commands, while the application development engineer can add required interlocks (manual disable, open disable, close disable). The control block accepts commands from both the HMI operator (cmode, copen, cclose—through VDS) and programmatic commands from automatic sequences (aut_req, open_req, close_req). The initial operating mode of the MOV is automatic. If the user needs to operate the MOV, he first has to switch the MOV into manual operating mode. This can be done on the faceplate for every MOV. Faceplates are created as one graphical block and the configuration for every specific MOV is done by linking adequate variables from the control blocks.

The faceplate that enables the operator to control and monitor the MOVs from HMI screens is shown on Fig. 7 and next to it is the external view of the functional block structure.

The faceplate consists of start/stop/close/esd (emergency shutdown) commands and a local/remote indication, which cannot be changed by the operator from the control room, but only from the local switch on the actuator itself. Automatic/manual command and indication of existence

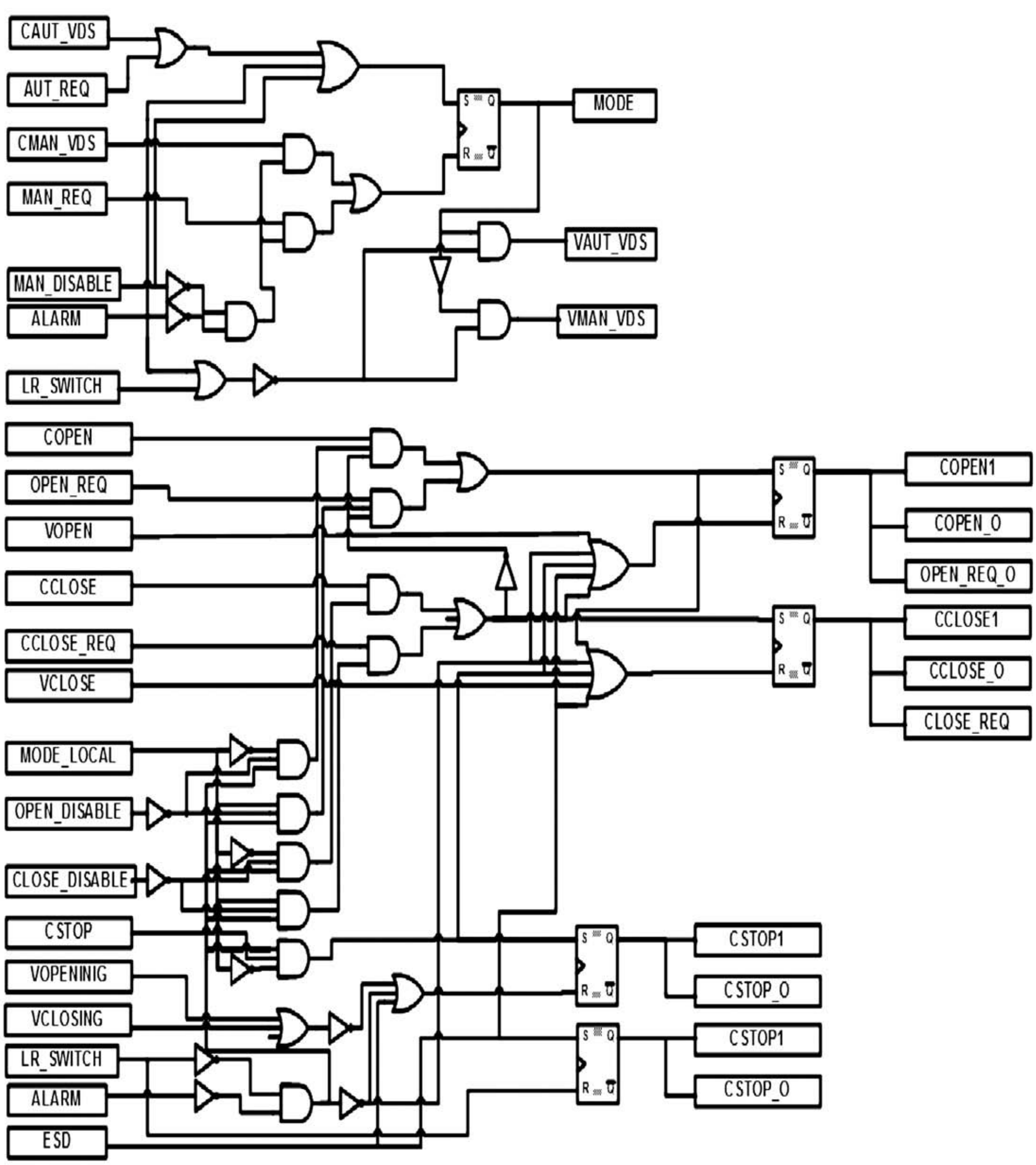

Fig. 6. MOV control block logic.

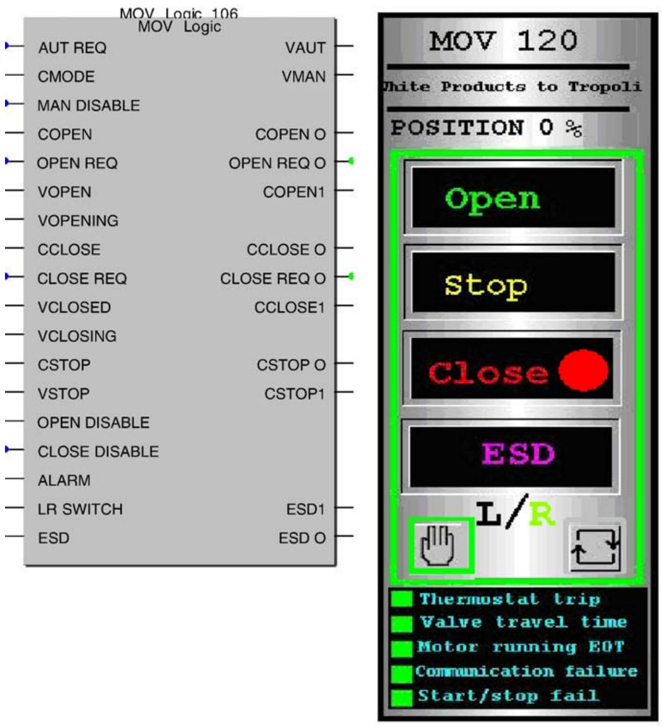

Fig. 7. MOV faceplate.

of the alarms and communication errors are displayed at the bottom of each faceplate [23].

2.5. User-defined block for pump control

The structure of the pump control block is shown on Fig. 8, followed by the picture on Fig. 9 displaying its external wiring connections and faceplate design for block monitoring and control.

As it is the case with the MOV control block, this block also has automatic/manual and start/stop logic, for commands issued both from human operator (manual control), and programmatically by control sequence logic (automatic control). The initial operating mode is always in automatic. Regardless, of the current operating mode, if the abnormal state occurs (low suction pressure, high discharge pressure), the operational mode immediately switches to automatic, and issues the command to stop the pump.

The pumps in Pump station A are programmed to work all the time in automatic mode, and cannot be switched to manual mode by the remote operator. The control logic is implemented in such a way that the pump starts every time the loading request is received from the loading gantry, and stops when the loading has finished. If more than one pump is present on the same product line, the starting of the second pump is implemented if the requests are received from multiple loading arms. Also, the automatic starting of the third (standby) pump is carried out, when one of the main pumps drops out of service.

For Pump station B, different control algorithms were implemented. The operator can start the pump manually, but only if the suction MOV of the pump is open, and the discharge MOV of the pump is closed.

The faceplate of the block consists of the start/stop and automatic/manual indications and commands, and local/remote status indications. Here again, the bottom of the faceplate is reserved to indicate all alarms and abnormal conditions related to equipment associated with the pump.

2.6. Control loops in the system

There are six central control loops implemented at the refinery Terminal, two for temperature control, and four for pressure control. Two heat exchangers, E-151 and E-152 are used to preheat the fuel oil before sending it to the pipeline in order to decrease the viscosity of the product and hence reduce the pumping power needed to transfer the fluid to its destination point. For both fuel oil and white products, pressure control is implemented in order to control the pipeline's delivery pressure. Control loops are of the cascade structure with the primary set point being corrected by product density and the temperature on the incoming side.

3. Example: LPG transfer through pipeline

Fig. 10, displays an operator screen for control and monitoring of LPG transfer to two remote locations through pipeline. The screen depicts two LPG spheres used as storage tanks for LPG at the terminal. Level measurement inside LPG spheres is done by using radar level transmitters, indicated on the screen as tags LI-108 and LI-109 and with graphic progress bars shown within spherical icons. Besides level indication, the screen also has pressure indication (vapor pressure at the top of the sphere) PT-104 and PT-105 as well as temperature indication TI-108 and TI-109. Each sphere has three level alarm indicators, which become visible as blinking dots when a set alarm level has been reached. Among all alarms, only LAH-108B is not included in the system control logic. This alarm represents an indication for the operator that the product level is high inside the sphere. LAHH-108A, which indicates a very high level of product in sphere S-108 is linked with MOV-138, which is the inlet MOV for sphere S-108. Occurrence of this alarm sets MOV 138 into automatic mode and control logic issues a close command to close the MOV in order to stop further level increase. This action has higher priority than any other action generated by the operator or automatic program sequence. In the same way, alarm LAL-108C is linked with the MOV-601 at the outlet of the sphere.

Occurrence of this alarm sets the MOV to automatic mode and issues a close command as a safety action for pump operation (every pump has also low suction pressure switch on its suction side, and control logic ensures automatic pump shutdown in case of low suction pressure alarm and trip).

In the same way, level indication, pressure indication, temperature indication and level alarms are implemented for sphere S-109 and also for control logic and interlocks and trips, as mentioned above.

From the screen an Operator can control all MOVs and pumps for LPG shipping to harbor for oversea tanker transport. Pump P-103 is a booster pump, which increases product pressure in front of the main shipping pumps P-104 and P-105.

If the operator wants to send LPG through one main pump (second is always on standby), booster pump P-103 needs to be on. As mentioned before, the pump cannot be started if the discharge MOV is not closed and the suction MOV is opened.

After the pump starts and the discharge pressure reaches a certain value (e.g. timer elapses), the control logic issues a command to open the discharge MOV.

The screen also shows pumps P-453A and P-453B, which can be used for LPG shipping to a New Terminal (located approximately 2 km from this terminal).

Employing the other type of status and availability indication, each pump is equipped with a low-pressure switch at the suction side (or differential pressure switch for pumps that take the product directly from spheres—Pumps 103, 453A and 453B). These signals are incorporated within control logic and in the automatic pump shut-down sequence. Apart from the above-mentioned control functions, the disable pump start function is also implemented at the low pressure at suction side. All of these interlocks and blocking functions are easily implemented in the software by just connecting specific variables to adequate function block inputs (alarm, man_disable, start_disable, aut_req, stop_req) of the previously described generic pump control block. In this way, the control logic development has been greatly simplified and accelerated.

4. Overview of the system multilayered structure and protocols in Project

4.1. System layer 1: proprietary field buses

As seen on the overall system topology depicted on Fig. 5, at the lowest level of HMI/SCADA system, apart from direct I/O field connections, there are three fieldbus subsystems: level gauging subsystem, MOV (Motor Operated Valve) subsystem, and truck-loading computer subsystem. The majority of I/O signals have been entered into the system directly through the PLC's I/O modules (around 800) located in the control room and the power transformer substation (mostly due to the hazardous area limitations). Nevertheless, three fieldbuses have significantly reduced overall wiring expenses, and enabled a much greater number of signals from connected devices to be collected into a central database. This has enabled not only signal processing but also online diagnostics and equipment troubleshooting [4,8].

4.2. Level gauging subsystem

The level gauging subsystem is comprised of the level gauging transmitters connected to a higher layer of the system with an Enraf bi-phase mark modulated fieldbus.

Level gauging instruments, like 970 SmartRadar ATi [19], measure and transmit the data upon request of higher layered system. They do so via a dedicated communication unit (CIU—Communication Interface Unit) as seen on Fig. 11. The Enraf fieldbus is the information carrier for a proprietary Master–Slave protocol, used for communication between CIU Prime and remote tank level gauging systems [20].

A maximum of 50 level gauges can be connected to the CIU Prime (15 gauges per Field Port), leaving us with enough reserves for possible future additions of new tanks. The Enraf fieldbus signal is not polarity-sensitive, and from experience gathered on many site installations, the best results were obtained when the Enraf fieldbus signal to gauges is connected in a "star" configuration, as depicted on the following Fig. 12.

The protocol between the communication interface unit (CIU) and the PLC in cabinet 2 is Modbus RTU, which is then converted to

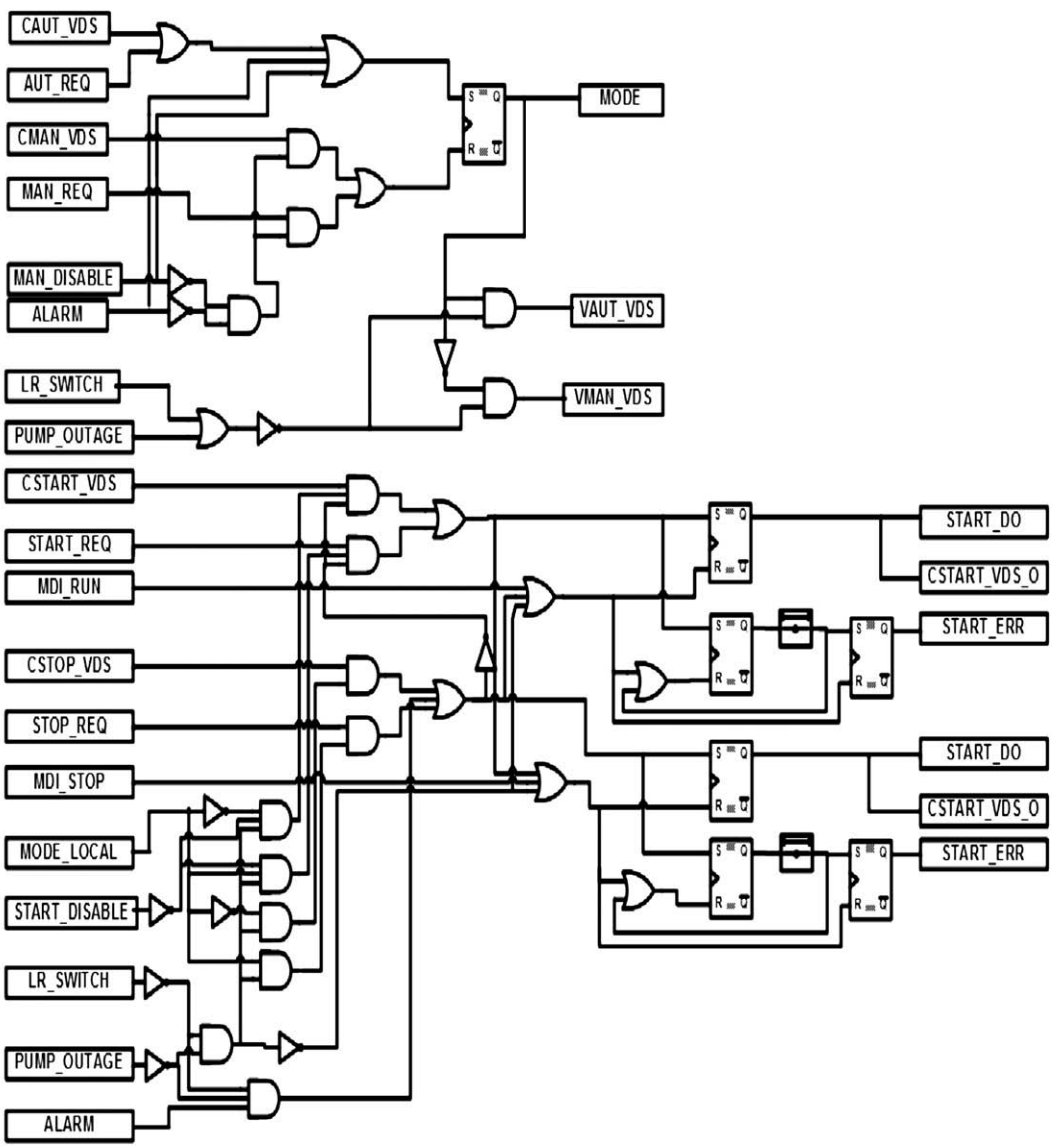

Fig. 8. Pump control block logic.

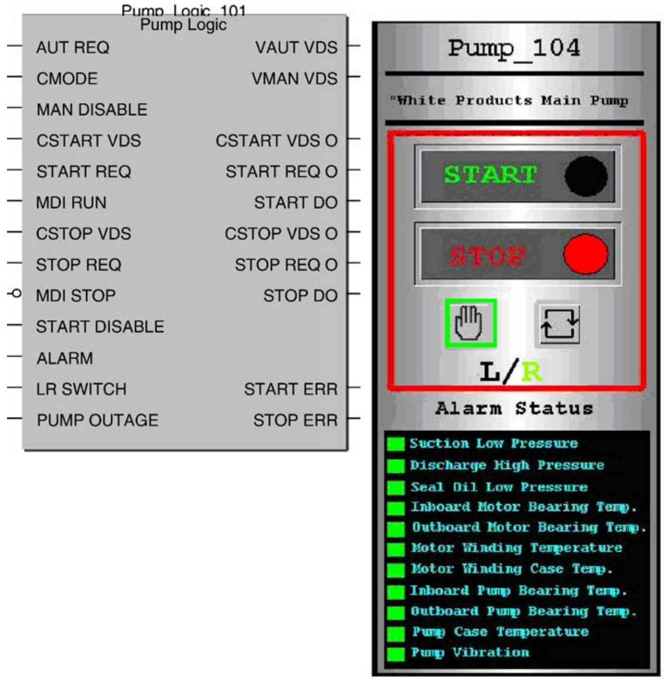

Fig. 9. Pump faceplate.

Modbus/TCP using Modbus RTU – serial to Modbus/TCP – Ethernet converter [24], inserted between the CIU unit and Ethernet switch (see Fig. 5 on overall system topology).

4.3. MOV fieldbus subsystem

The MOV communication subsystem consists of the three basic components: A Pakscan master station, which is in full redundant configuration, a Pakscan fieldbus and motor, operated valve actuators [17].

4.3.1. Pakscan field bus

The Pakscan fieldbus provides the link between valve actuators and supervisory control. It is an intelligent, reliable, and high integrity proprietary network, specifically designed for use with Rotork Actuation products.

The communication model over the network is Master–Slave, so that the field unit may not transmit any data unless it receives a request from the master station. All data messages and commands are verified by framing with vertical parity check and longitudinal CRC checks.

Data exchange between the Pakscan master station and PLC in cabinet 1 is based on the Modbus protocol using Modbus/TCP physical level and protocol level to interconnect Pakscan redundant NIC cards to two independent Ethernet switches forming a fully redundant ring LAN network.

4.3.2. Pakscan loop wiring

The Pakscan network uses a twisted pair cable that carries a 15 V, 20 mA current loop signal. The signal is modulated by the Pakscan master station to send and receive data from the attached field units. The use of a 20 mA current loop as a digital signal carrier automatically ensures that the system offers low impedance to any electromagnetically induced noisy currents and prevents these currents from generating significant voltage spikes in the signal transmission loop.

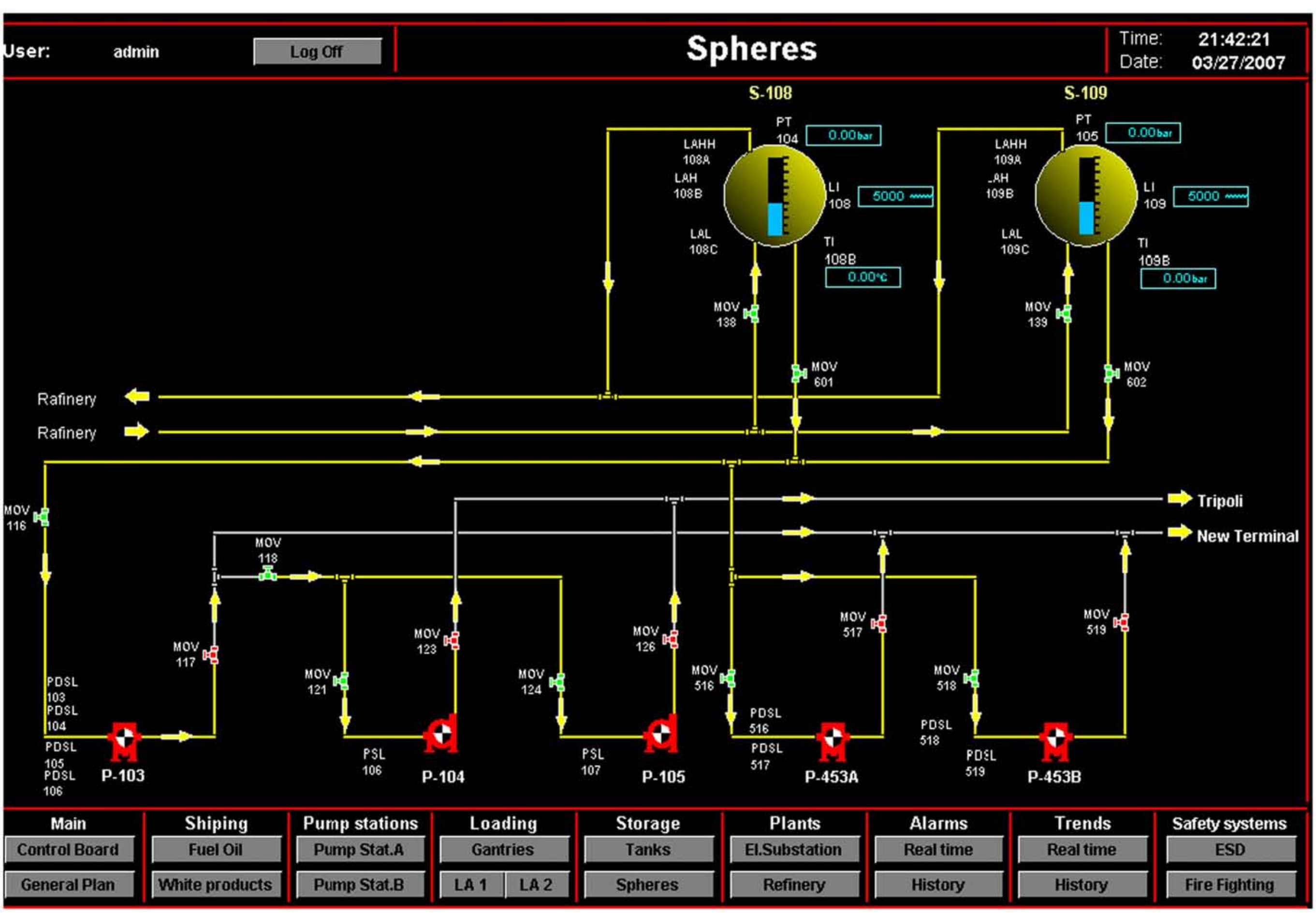

Fig. 10. Spheres screen layout.

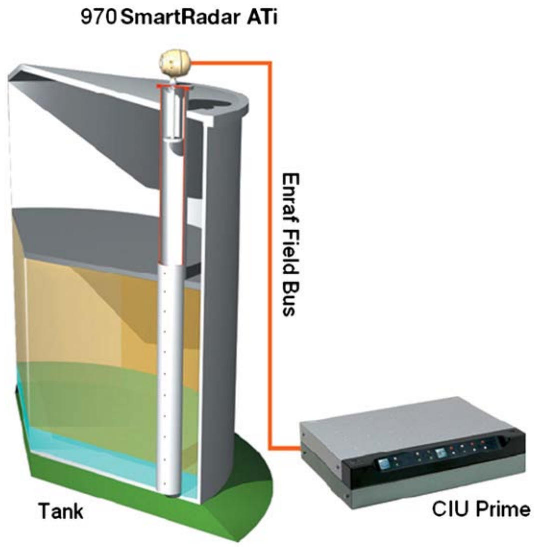

Fig. 11. 970 Smart radar level gauge.

At the master station side, full galvanic isolation is maintained between the two-wire loop connections and the processors. Each field control unit is fitted with a micro-processor, an EEPROM, to hold the address and communication speed, and a detector to sense the loop current. As with the master station, a field control unit maintains full galvanic isolation between the loop signal detection circuits and actuator electronics.

4.3.3. Pakscan network fault tolerance

Two wires are connected to, and taken from, each field control unit in turn. They originate from and return to the master station to create single twisted pair, 2-wire loop.

As each device may be accessed from either direction, a redundant communication path is available (see Fig. 13). Pakscan fully utilizes this fact in the event of cable fault (open circuit, short circuits and ground fault).

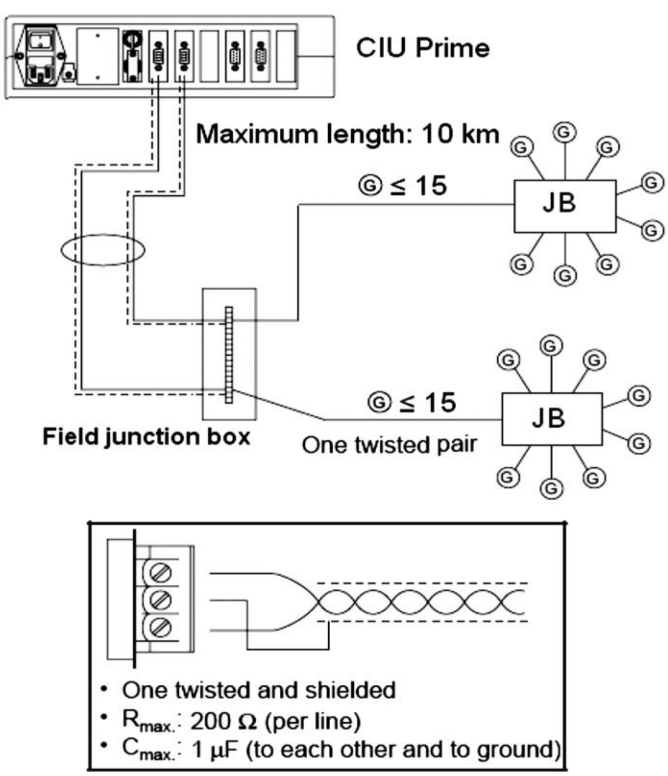

Fig. 12. Connection of radars to CIU unit.

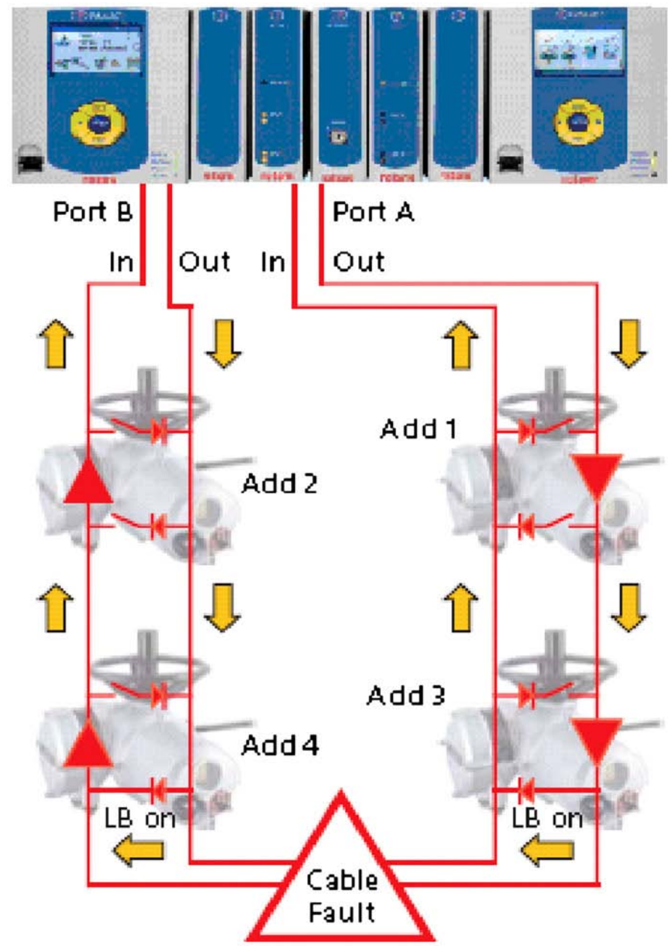

Fig. 13. Redundant connection of MOVs to central unit.

4.4. Truck-loading system

The truck-loading system is composed of a flow meter, strainer and air purger, set-stop valve, temperature probe and a card reader all connected to an embedded type microcontroller device in role of

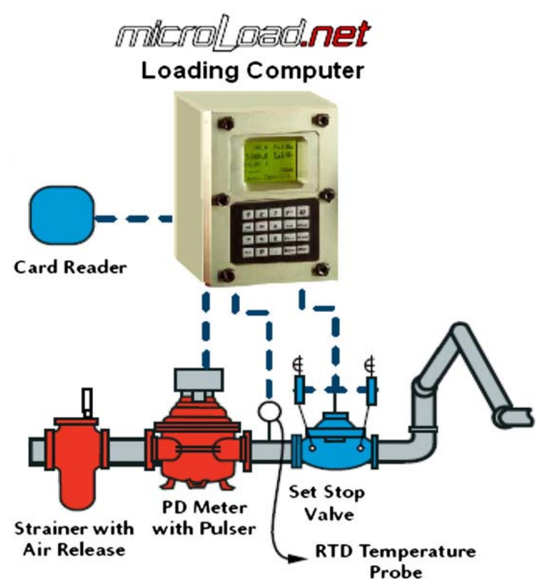

Fig. 14. Truck loading metering set.

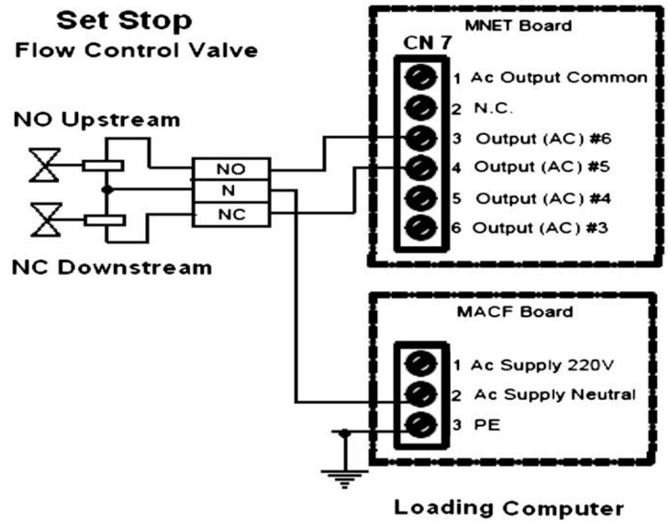

Fig. 15. Set-stop valve interconnection.

loading the computer which displays delivery information, and controls the loading process. It is depicted on Fig. 14.

Load computers [25] are interconnected redundantly (over two serial ports) using a three-wire connection of a multidrop serial RS-485 bus to PLC in Cab1 having two channel redundant RS-485 serial cards. Since the FMC/Smith Load computer supports neither the Modbus protocol nor the automatic switchover on two serial ports, a large amount of work and time had to be invested to develop a protocol driver within the PLC logic designer and data server in order to exchange a significant amount of data and commands between the load computers, the PLCs and the data server.

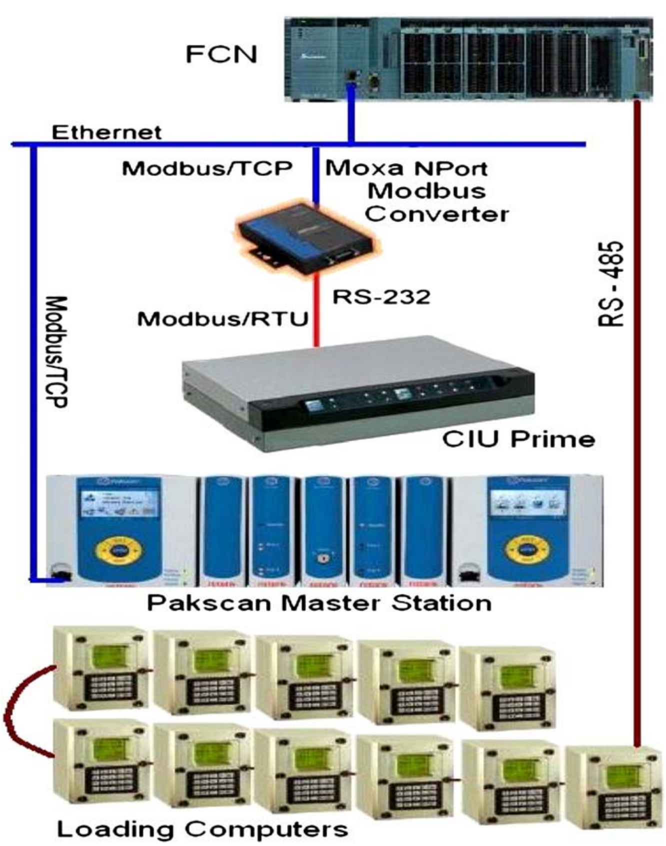

Fig. 16. Flow computers and MOVs connection to PLC.

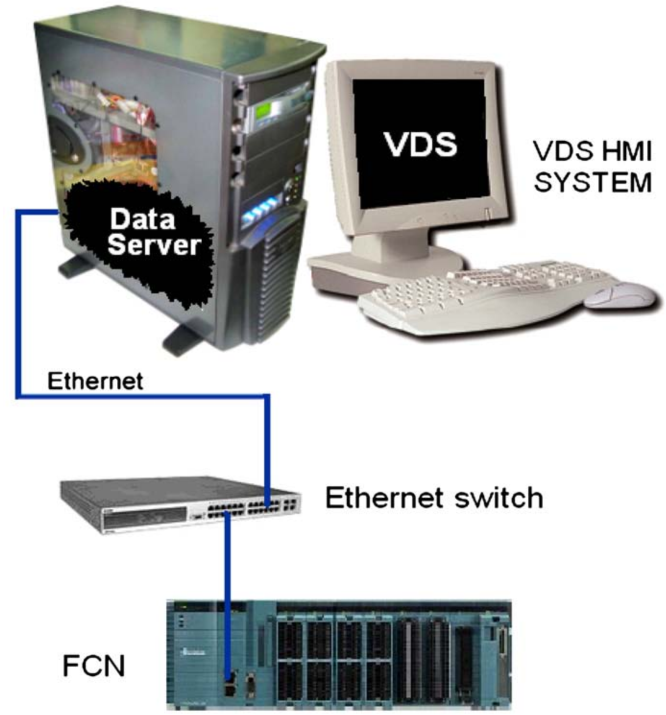

Fig. 17. Data server- PLC interconnection.

4.4.1. PD flow meter

PD (Positive Displacement) meter has a built-in dual pulse transmitter and is connected to the loading computer. On the loading computer the K-factor is set for relationship between pulses and engineering delivery unit (13 pulses = 1 l , K = 13).

The RTD (Resistances Temperature Detector) supplies the loading computer with the resistance value of the sensor immersed in the product, from which the temperature can be calculated, and volume flow metered by PD meter corrected for actual loading temperature and mass flow and loaded mass quantity calculated.

4.4.2. Set Stop valve

The Set Stop valve is a series valve with two solenoid controls. The normally-open (NO) and normally-closed (NC) solenoids, located in the upstream and downstream portions of the product flow line, respectively, control the operation of the valve. The Fig. 15 shows the way the set-stop valve is connected to the loading computer and controls the flow.

4.4.3. RF-ID card reader

Card Reader is an RF-based proximity reader with high performance and reliable method of identifying truck drivers and loading users [26]. Card Reader is capable of interpreting multiple card formats and transmitting card data to the load computer.

The ID of the truck driver received from the card reader is used by the load computer to check in its online database whether the driver can be allowed to start the loading related to his company's financial status, as well as to issue the loading bill and register the loading quantities in the central billing and invoicing database located at the Company headquarter.

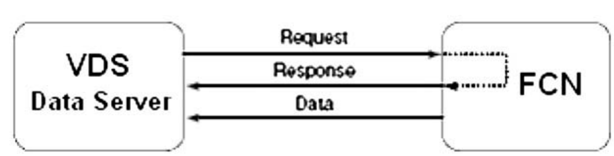

Fig. 18. Data server PLC message exchange.

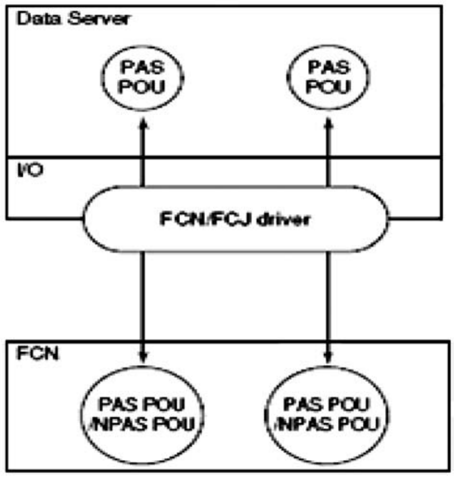

Fig. 19. Data flow between PLC and data server.

These two: process and business databases, are online interconnected and synchronized over the Internet using OPC protocol tunneling within VPN network with Matrikon OPC to ODBC converter [27].

4.5. Layer 2: PLC—fieldbuses communication

The connection of all of the three fieldbuses is depicted on Fig. 16. Two of them have been adapted for talking Modbus (either RTU or TCP) to PLCs, while in the case of loading computers; a specific protocol driver was developed.

4.6. Layer 3: PLC—HMI communication

In this layer, a data server polls all connected controllers at a user-defined polling rate, which is different for different parameters. The controllers are usually set as passive type devices so they just pass the requested data to the data servers. Some of them are set as active type devices, and can transmit data unsolicited, at an arbitrary timing rate [28] (Fig. 17).

FCN (Field Control Node) as PLC, does both, it provides response to a request from the Data Server, but also supports unsolicited data

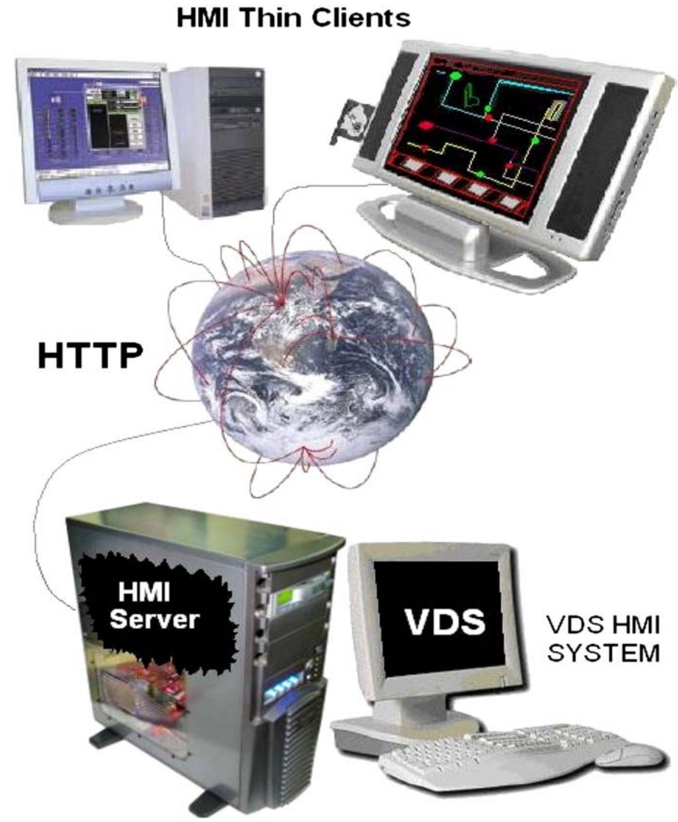

Fig. 21. Web server–thin client interconnection.

transfers, which means it is set as a hybrid type device as depicted on the following Fig. 18.

Careful tuning is needed, when connecting a PLC to a VDS Data Server, in order to register virtual IP addresses of an FCN, set whether or not a dual network will be used, and whether or not the time between the FCN and VDS should be synchronized.

4.7. I/O objects

Data server is composed of the data server engine, Control objects and the I/O objects. The data server engine exchanges data with the field devices using I/O object function, and executes various processing using control objects.

The I/O object provides the following functions:

- Input/output of the device data from the controller
- Asynchronous event reception from the controller
- Specific control of the controller

4.7.1. Importing application data lists

As mentioned earlier, Logic Designer is a set of programming languages for writing a control application that will run on the FCN.

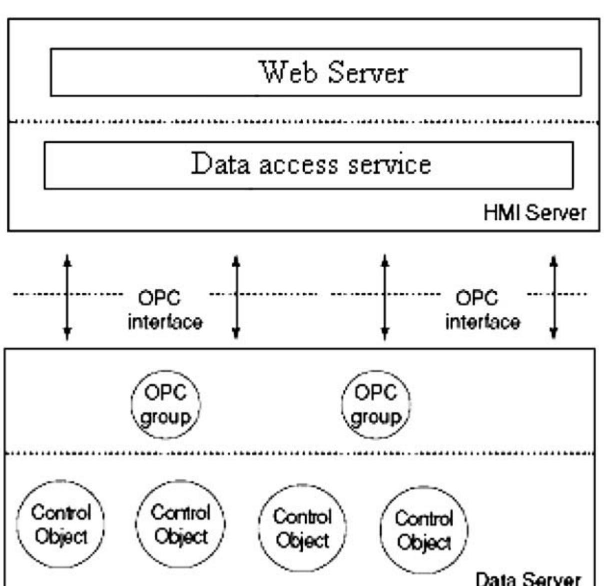

Fig. 20. Data flow between data server and web server.

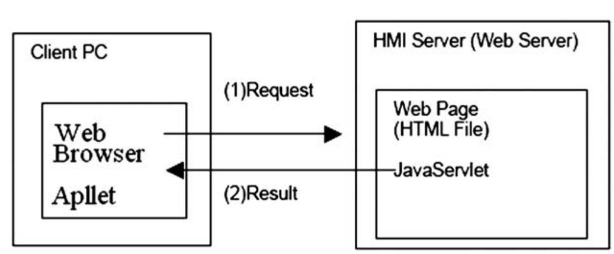

Fig. 22. HMI server-thin client data exchange.

When the control application is compiled, Logic Designer saves the program code organized in program organization units (POU) and definition information, as autonomous controller/application data lists (ADLST).

When connecting with an autonomous controller FCN, Control objects use POU objects. Control objects names are the label names for the POUs.

Related I/O objects and POU objects in Data Server corresponding to the POU in FCN are defined automatically when Object Builder is used to import autonomous controller/application data lists (ADLST), as shown on the following Fig. 19.

POU objects are imported as a single group for each controller. The name of the group is the same as the name of the controller. (Resource name for Logic Designer, in the designed Project were: FCN001 and FCN002).

4.8. Layer 4: OPC communication

OPC client applications can be used on the same computer as the FCN OPC server or on any computer connected via Ethernet. As previously described, it was also used to exchange data between a data server and an Oracle database as OPC client over Internet

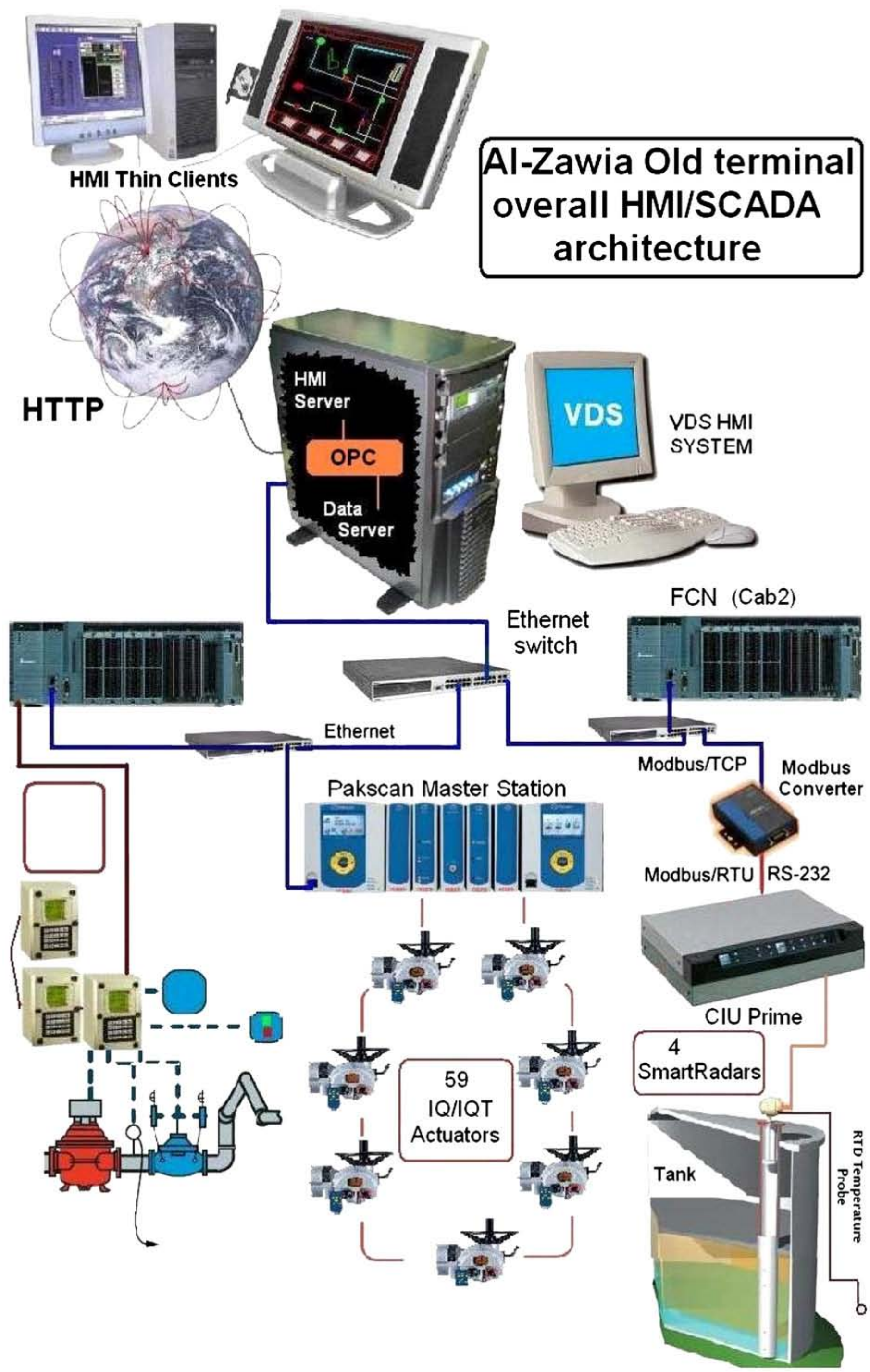

Fig. 23. Overall HMI/SCADA architecture.

VPN network [27]. The FCN OPC server offers access functions via OPC-DA for variable instances held by the control application. Access from OPC may be possible for data of the POUs offered by the portfolio, as is depicted on Fig. 20, showing that OPC communication is used between data server and any HMI server or client software, including VDS server. OPC-DA interface performs data conversion to the VARIANT type in order to match the OPC specifications that OPC clients are expecting to receive from the OPC server [29,30].

In VDS, Data Server is the OPC Server and HMI Server is the OPC Client, as shown on the Fig. 20.

4.9. Layer 5: HTTP

At the top layer, the HMI Server runs as message management software, serving as the HMI client's interface with Data Server, as shown on Fig. 21.

The HMI client is a standard Web browser that receives and displays data from a HMI Server. A special feature of the HMI client is that it is a thin client, meaning that no additional software is required for it to carry out the above functions, apart from a standard Web browser, significantly reducing the total cost of system integration. As a basic security feature, data access is limited depending on user privileges, and an add-on utility program called operation shield package. This utility is used to restrict access via keyboard commands to critical system operations such as system shutdown, and preventing Operator to misuse PC for other purposes except as dedicated Operator station. An additional feature of this utility package is to hold always-on-top foreground display of specified windows, as well as limiting the number of open windows. This feature turned out to be very useful, especially on the touch screen Operator's panels, were the pop-up windows, called by selection of control actions on faceplates, tended to hide and accumulate behind the main screen when losing the focus.

Attempts were made to change the operational mode of these pop-up windows and hold them up on the top of the other windows, until the Operator closes them by touching the appropriate control area of the window. This has proved to be not possible due to the minimal modification that applets could do on the client machine to change the type of the window in which they are executed by Java runtime engine (JRE) in a sandbox-like environment.

HMI graphic software displays a page, based on the same mechanism as is done over the Internet. Web browser specifies the address of the page to be displayed and issues the request to an HMI server (1). The HMI server returns the HTML file of the requested page (2).

Furthermore, the HMI server executes a program (JavaServlet) on it, and then returns the results to the browser in the way depicted on Fig. 22.

The following Fig. 23 illustrates data flow from three fieldbus networks to the two backbone PLCs representing a part of the SCADA system (direct I/O modules are not shown). VDS server and web based thin clients representing the HMI part of the overall system, implemented in the described refinery Project are also shown.

5. Conclusions

The paper describes and illustrates the application of a standard SCADA/HMI system, which has been significantly customized and extended through its open hardware and software architecture. There are many advantages of the described network based multilayered system. These are: modularity and extensibility, and the communication with the remote devices of various manufacturers and vendors, This is done by using Ethernet, OPC or any type of fieldbuses based on serial communication and myriad of protocols including tools for user specified communication and protocol driver. Additionally, the dual-redundancy architecture for the majority of the system components, and the modular software architecture with the standard and the open real time operating system are also valuable assets.

Number of functions are incorporated in the software tools, especially the layering capabilities for software development in the Logic Designer. These possibilities are demonstrated through control logic development and user-defined blocks developed for an industrial application. Within the HMI interface, the modern approach of Java based web application with thin clients further extends the modularity and scalability features of the system.

The described SCADA/HMI system integration approach, based on standard and of the shelf hardware and software modules and components, together with avoidance of any proprietary traps, has proved to be the optimal solution for system integrators of monitoring and control systems. This is especially true when these systems are extending their boundaries of real time data acquisition and processing to a higher level of plant, corporate and enterprise management and data processing, integrating process control and business activities in total plant and enterprise wide systems. Holding the integrated systems solution open, both hardware and software wise, gives the system integrator total freedom in solution selection.

References

[1] Yokogawa Stardom system documentation, Japan.
[2] K.J. Aström, B. Wittenmark, Computer Controlled Systems, Prentice Hall, London, 1997.
[3] M. Blanke, et al., Fault-tolerant control of complex systems, Control of Complex Systems, Springer Verlag, September 2000.
[4] G. Wood, Fieldbus and intelligent devices, Editorial, Fieldbus Special Issue, IEE Journal of Computing and Control 6 (1995).
[5] U. Lutze, Reduce control system complexity with fieldbus technology, Control Engineering 42 (12) (November 1995) USA.
[6] IEEE Std 999-1992 Recommended Practice for master/remote supervisory control.
[7] API Recomended Practice 557, Guide To Advanced Control Systems.
[8] H. Schumny, Fieldbuses in measurements and control, Computer Standards & Interfaces 19 (Issue 5–6) (October 1998).
[9] G. Grygiel, O. Hensler, K. Rehlich, DESY, DOOCS: A Distributed Object Oriented Control System on PC's and Workstations, , 1996.
[10] Maarten Koning, Keith Wiles, Device Software Optimization for Concurrent and Consecutive Systems, White Paper, Wind River Systems Inc, 2007-07-26.
[11] Zoltan Laszlo, Memory Allocation in VxWorks 6.0, White Paper, Wind River Systems Inc, 2005.
[12] B. Wittenmark, B. Bastian, J. Nilsson, Analysis of time delays in synchronous and asynchronous control loops, Proc. of 37th IEEE Conference on Decision and Control, Tampa, USA, December 12–15 1998.
[13] N.P. Mahalik, Attributes of industrial machine and process control systems, International Journal of Computer Applications in Technology (IJCAT) 25 (4) (2006).
[14] Ivica Crnkovic, Christine Hofmeister, Ralf Reussner, Quality of Software Architectures, Second International Conference on Quality of Software Architectures, QoSA, Springer Berlin, Heidelberg, 2006 ISBN-10: 3-540-48819-7.
[15] W. Heinzelman, A. Murphy, H. Carvalho, M. Perillo, Middleware to support sensor network applications, IEEE Network Magazine 18 (2004).
[16] API RP 554, Process Instrumentation and Control.
[17] Pakscan II E, high integrity master station. Rotork Publication S110E, issue 09/04.
[18] N.P. Mahalik, S.K. Lee, Some Studies on Designing Distributed Intelligent Actuating Systems for Industrial Control Applications, 85, Institution of Engineers, India, May 2004.
[19] 970 Smart radar Ati—Instruction manual, Enraf BV publication, Issue Feb. 2004.
[20] Instruction manual series 880 CIU prime, Enraf BV publication, Issue Nov. 2003.
[21] P.B. Gibbons, B. Karp, Y. Ke, S. Nath, S. Seshan, IrisNet: An Architecture for a Worldwide Sensor Web, IEEE Pervasive Computing, vol. 2, 2003.
[22] O. Hensler, K. Rehlich, DOOCS: A distributed object oriented control system, Proceedings of XV Workshop on Charged Particle Accelerators, Protvino, 1996.
[23] N.P. Mahalik, S.K. Lee, A study on production line automation with LonWorks control systems, Computer Standards & Interfaces 24 (Issue 1) (March 2002).
[24] Nport 6110 Modbus/TCP to serial cummunication gateway —User Manual, Moxa Technologies Co, Issue March 2003.
[25] Smith meter microload.net—Cummunication, FMC Technologies publication, Issue 10/2005.
[26] Smith meter proximity card reader—Operation. FMC technologies publication, Issue 12/2005.
[27] Matrikon OPC Server for GDA—user manual, Matrikon publication, Vers. 1.2, December 2005.
[28] N.P. Mahalik, J. Ryu, B.H. Ahn, Simulation integrated management layer for real-time embedded DCN, Computer Standards & Interfaces 28 (Issue 5) (June 2006).
[29] OPC—Data access automation interface standard, OPC Foundation publication, Ver. 2.20, Feb 1999.
[30] OPC—Data access custom interface standard, OPC Foundation publication, Ver. 3.00, March 2003.

Prof. Adnan Salihbegovic is a full time professor at the Faculty of Electrical Engineering, Department of Computing and Informatics and Department of Control Systems and Electronics, University of Sarajevo, Bosnia Herzegovina. He received his diplome engineer degree at the Faculty of EE in Sarajevo (1966), Master degree at UMIST in Manchester, England (1967), and Doctor's degree from the University of Sarajevo (1985). He has been working as researcher and research director for 25 years at the Institute for Automatic Control and Computer Sciences of Energoinvest Company in Sarajevo (1966–1992), as well as chief control and computer systems design engineer at the Ras Lanuf oil refinery in Libya (1993–2003). During his 40 years of academic and professional career he has published and authored more than 130 scientific, technical papers and study reports and has received many awards for his achievements. He is the holder of 5 patents. Prof. Salihbegovic is a member of ISA, ACM and IEEE societies. His major current research interests include modeling and simulation, real time and embedded control systems, real time operating systems, process and business database data mining and integration, and networking protocols and fieldbuses.

Vlatko Marinković is currently working as control systems design engineer for Bosna-S Oil Services Company in Sarajevo. He received his diplome engineer degree in Electrical Engineering at the University of Sarajevo in 2006. During his study at the Faculty of EE, he actively participated in several research and design projects as research student. He is currently a M.Sc student of the same University, Department of Automatic Control and Electronics. His research and design interests are in SCADA/HMI Systems, artificial intelligence, modeling and simulation, PLC programming and Oil & Gas measurement and control design solutions.

Zoran Cico received his diplome engineer degree from the Faculty of Electrical Engineering at the University of Sarajevo in 2007. During his study at the Faculty of EE, he actively participated in several research and design projects as research student. He is currently a M.Sc student of the same University, Department of Computing and Informatics and part time teaching assistant of the same department. He is also employed as system integration engineer for Energoinvest RTC-AMT Research Institute in Sarajevo. His research interests are in SCADA/HMI systems, GIS, industrial networks and protocols and monitoring and control of small unmanned hydroelectric plants.

Karavdić Elvedin is currently working as design engineer for Bosna-S Oil Services Company in Sarajevo. He received his diplome engineer degree from the Faculty of Electrical Engineering at the University of Sarajevo in 2007, Department of Automatic Control and Electronics. During his study at the Faculty of EE, he actively participated in several research and design projects as research student. His research and design interests rest in SCADA/HMI Systems, modeling and simulation, PLC programming, electronics and microcontroller programming and automation in Oil & Gas industry.

Nina Delic received her diplome engineer degree from the Faculty of Electrical Engineering at the University of Sarajevo in 2007. She is currently working as a director of Briefing Media Intelligence Company. During her study at the Faculty of EE, she actively participated in several research and design projects as research student. Her research and design interests are in SCADA/HMI systems, industrial networks and protocols, business intelligence and data mining.