

SOFTWARE SCENARIO FOR CONTROL SYSTEM OF INDUS-2

R. K. Agrawal*, Amit Chauhan, A. M. Gupta, B. Merh, P. Fatnani, S. Gangopadhyay
 Centre For Advanced Technology (CAT), Indore, India

ABSTRACT

This paper describes different software technologies used to control Indus-2 machine and its subsystems. This includes communication protocols between different layers of Indus-2 control system, operator interface to machine, web-interface to machine data and machine application programs.

INTRODUCTION

Indus-2, a 2.5 GeV Synchrotron Radiation Source (SRS) under construction at CAT is having three-layer control system architecture [1]. Software technologies and software model are based on the distributed client-server architecture, employing object-oriented methodology and commercial Supervisory Control And Data Acquisition (SCADA) package PVSS-II® [2], at the top layer. The software model is based on open architecture, which provides flexibility in interfacing front-end instruments with varied standards and adaptability to fast changing hardware and software technologies and optimal use of available resources. This open and flexible architecture supporting multiple software technologies is an important requirement as the accelerator machine employs multi-disciplinary engineering having long term, diversified and ever-changing requirements.

The controls system of INDUS-2 has three-layered architecture [1]. Layer-1 is User Interface Layer (UI) has PC workstations for GUI. Layer-2 is Supervisory Control Layer (SC) has VME stations for supervisory control and Layer-3 is Equipment controller Layer (EC) has VME stations that house various I/O boards. Communication between SC and EC Layers uses Profi field-bus protocol [3].

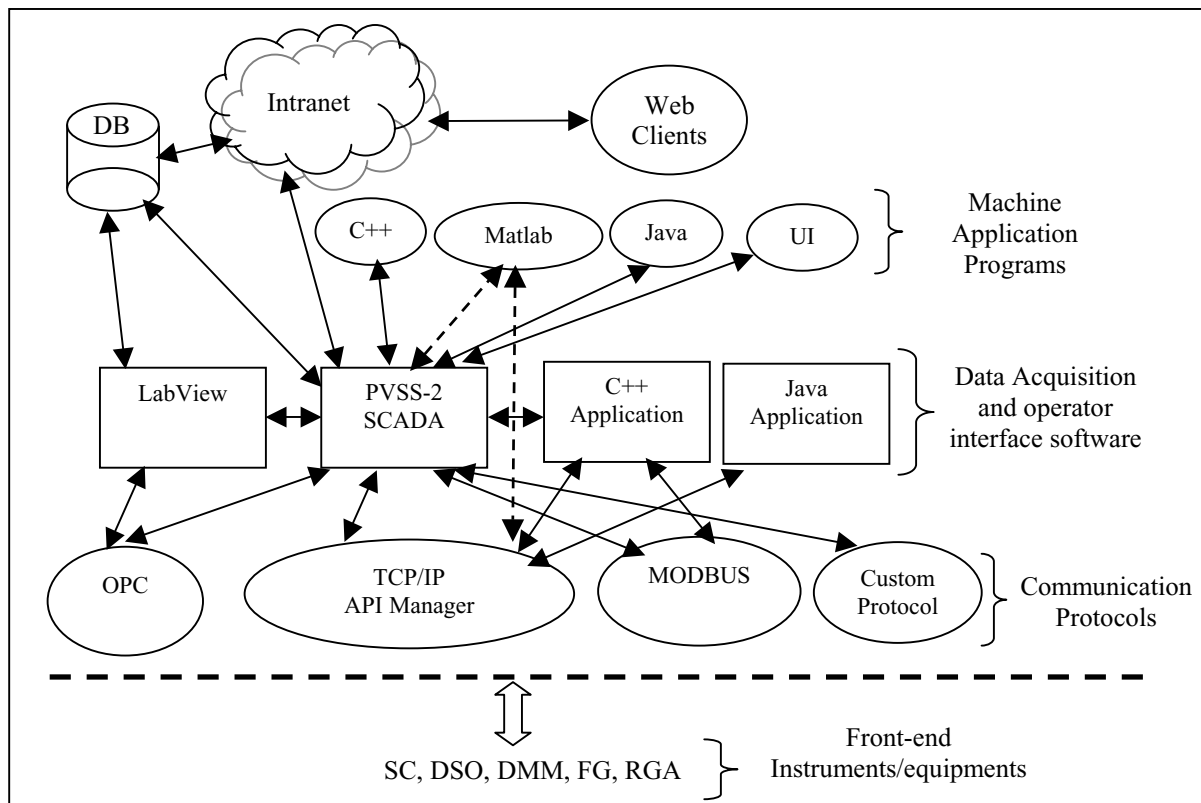


Figure 1: Software Architecture for Indus-2 Control System

*ragrawal@cat.ernet.in

COMMUNICATION PROTOCOLS AT UI LAYER

The front-end instruments and equipments to be interfaced to the control system are Supervisory Controllers (SC), Digital Storage Oscilloscope (DSO), Digital multi-meter (DMM), Function Generators (FG), Residual Gas Analyzer (RGA), spectrum analyzers and custom power supplies etc. Normally they have different communication standards like GPIB, OLE for Process Control (OPC), TCP/IP custom socket communication, MODBUS etc. All these standards are supported and any new or custom protocol can also be easily added in the software architecture. Use of industry standard protocols like OPC and MODBUS provides seamless interface of commercial instruments. For specialized instruments OPC servers or any other custom server like socket server can be developed so that they are independent of data acquisition and operator-interface applications.

SOFTWARE TECHNOLOGIES AT USER INTERFACE LAYER

Indus-2 control system uses PVSS-2 SCADA at the UI layer as the main component. This is having an object oriented data-point concept and thus common objects can be identified and developed as reusable components. It provides interface to standard protocols and it can even be extended to custom protocols. Use of commercial SCADA package provides faster and better development and ease of expansion in the future. Data trending, Alarms, configurations, and data history are easily implemented with the help of PVSS. LabView[®] interface application, custom C++ applications and Java applications can also be used to handle the data coming from the front-end instruments. All these applications (clients) take data from the server applications supporting different protocols. Thus with this client-server architecture multiple applications can simultaneously acquire the data. This modular and client-server architecture decouples the GUI from the field related I/Os. PVSS uses API manager to interface to the SC layer server application. These Application Programming Interface (API) managers have filtering to reduce the number of events sent to the PVSS Event Manager (EV).

API Manager

API Managers are used to extend the functionality of the PVSS SCADA in order to interface it to the SC layer of the Indus-2 control system. The API managers communicate to the SC layer server applications on Ethernet over TCP/IP socket. They poll the SC layer server applications periodically to get the current state of the sub-system. User commands and settings are given in the PVSS User Interface Manager (UIM), which in turn are transferred from API manager to the SC layer without wait. Each sub-system of Indus-2 has a SC layer and corresponding API manager. API managers have various features e.g. auto connects to the SC layer server and provides maintenance and testing related information to the GUI.

OPC Server

OPC servers for various DSOs are developed to monitor various signals on the desktop. OPC being an industrial standard, monitoring and control of these DSOs are possible with various software packages e.g. PVSS, LabView and custom OPC clients. The Publish and subscribe model of OPC is very useful for getting signals whenever they change.

SOFTWARE TECHNOLOGIES FOR MACHINE APPLICATION PROGRAMS

MATLAB[®] is very well suited for mathematically intensive machine applications like closed-orbit correction, coupling correction and correction of Beta asymmetries. MATLAB[®] [4] provides a very good Graphical User Interface Development Environment (GUIDE) and has inbuilt matrix and graphic functions which greatly reduce the development time. Its Accelerator Toolbox [5] and Linear Optics from Closed Orbit (LOCO) are being used for machine simulation and calibration.

These applications can take the online data from PVSS SCADA with the help of API managers implementing custom protocol over TCP/IP or directly from the server applications. These higher-level programs can also be implemented in C++ and Java and can perform the required calculations on the acquired data to generate modified machine settings which can be applied to the machine through the same API manager interface of the PVSS-2 or directly through the server applications. The PVSS

SCADA has to handle a large number of I/Os so it is quite possible that the fast response time as required by machine application programs will not be achievable. Direct writing to server applications with intimation to the PVSS SCADA can be considered for this type of requirements. (As shown in dotted line in the Figure.1)

SOFTWARE TECHNOLOGIES FOR WEB INTERFACE AND DATA LOGGING

The databases for Indus is designed to store data such as beam current, beam energy, status and read back values of various devices, configuration details, device set points, alarm settings etc. The data is also archived to provide important trace-back information regarding the physical behavior of the components. The database users can obtain the real-time and archived data through web browser from anywhere on the network.

The data from all the subsystems is logged periodically to a Database. The Database Management System (DBMS) manages mainly three categories of data. *Configuration database* stores static data of equipment like calibration constants, physical dimensions, configuration details etc. The range configurations defined for data depending on resolution, accuracy etc. is also maintained in the database. These details are used for data validation at the time of logging. The configuration details can be changed by updating the corresponding data in the database. *On-line database* stores the current status of the accelerator parameters. This data comprises of read-back values, remote/local and On/Off status of equipments. *History database* stores data periodically extracted from on-line database with date and time stamps. Along with these an Alarm Database is also being managed which logs all the alarms occurring in the system. A set of functions written in Structured Quarry Language (SQL), accept the period (start & end) and the equipment for which data is required, and return the same data as it had come from the logging application. This logging is done to a commercial database using the ADO/ODBC interface from the PVSS-2. The database is designed to handle the huge data-rate. Other Interface applications can also log the data to the database.

The data archiving and retrieving system in Indus is based on a client/sever model. The TCP server listens for client requests, accepts them, reads data from the VME and sends it back to the client. The client then logs it into the database server. Data acquisition process issues one SQL command to one table per one data acquisition cycle. Data duplicity is prevented using triggers.

Online machine data from the PVSS SCADA can be retrieved from the web-browsers. The on-line and off line data is presented to users with the help of web interface programs. For this web based data presentation applications are designed using the web technologies. Presently Beam and shift information is taken periodically from the database and displayed on different locations in the INDUS complex. We are presently using Internet Information server IIS as the web server and Java Server Pages (JSP), Active Server Pages (ASP) and Servlet for the server-side applications with Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) interface to the database.

SOFTWARE TECHNOLOGIES AT SC AND EC LAYER

SC Layer consists of various VME crates, each housed with a Motorola 68040 CPU board and a Profi bus Master card while EC Layer consists of various VME crates, each housed with a Motorola 68000 CPU board, a Profi bus Slave card and various Digital and Analog I/O cards.

All the CPU boards have Real Time Operating System (RTOS) OS-9 ported on them. The various OS-9 programs handle inter layer communications, accessing the I/Os and provide inter process communication. A socket server module at SC layer CPU handles the SC to UI layer communication and a process that implements the Profi Master Application Layer Protocol handles SC to EC layer communication. The EC layer CPUs have programs to implement Profi bus Slave Application Layer Protocol and programs and Device Drivers [6] to control various I/O boards.

An application program in OS-9 accesses I/O boards using the routines in File Manager and Device Driver modules [6] for that board. Device Drivers have been developed for the various boards and a general File Manager has been written which is used for all types of boards. In addition, Device Descriptor [6] modules have been made for every I/O board that contains specific information like its base address, IRQ level, IRQ vector etc.

At SC Layer there is networking support, so BootP/TFTP [6] protocol can be used there to download the OS-9 boot-file and application programs from the workstations at UI Layer at startup. This will

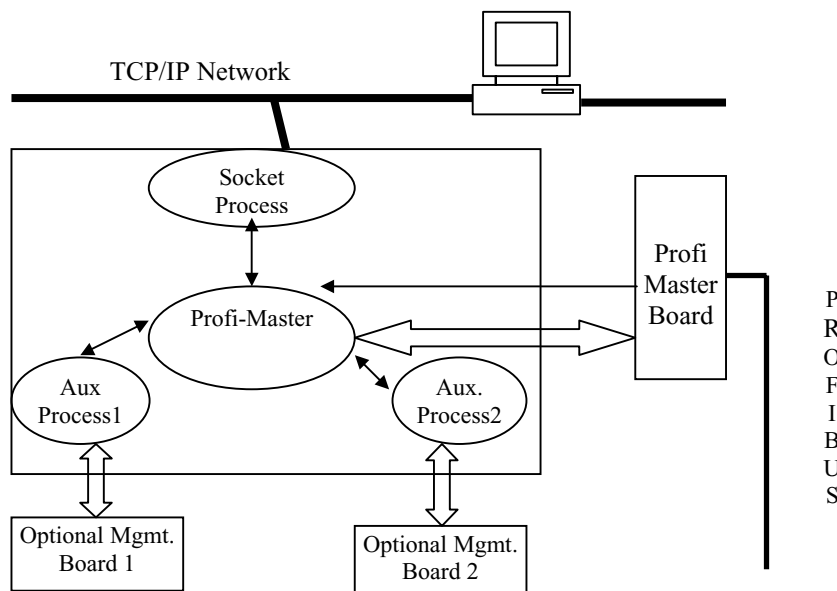


Figure 3: Scheme at SC layer

make it easier to modify and load the programs and data at SC Layer that also contains the parameterization information for EC Layer.

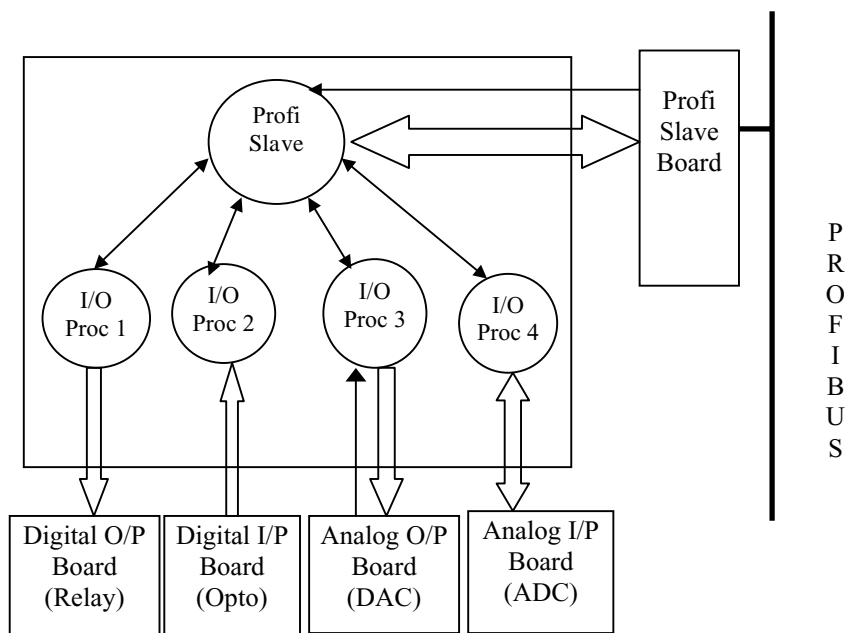


Figure 4: Scheme at EC layer

The use of OS-9 provides the facilities like multitasking, ease of programming, modularity and scalability, good inter-process communication features, easy maintenance and updating of programs, rommability and small footprint allows the use on embedded cards with limited memory. Above all, it guarantees time-bound response to external events like interrupts for some critical actions to be taken, which may prove to be useful in beam orbit correction schemes.

The programs to implement the Data Link layer of both ProfiBus Master and Slave runs as firmware on respective ProfiBus controller cards. These Profi controller cards are based on micro-controller controller DS 80C320/80520 running at 24 MHz.

The stations have a typical response time of 100 microseconds and the data communications over RS-485 link has 750 K baud rate.

PRESENT STATUS

Presently this system is catering to data acquisition and control for seven sub systems of Indus-2 that have in all seven SC stations and 75 EC stations spread in the field. This system is handling ~10,000 I/O points. Instruments like DSOs, Spectrum Analyzers, video monitors etc. are also interfaced with the control system. The various GUI panels, API managers and server programs at SC layer provide a comprehensive machine interface to the users.

CONCLUSION

The various softwares in the control system are providing the required functionalities. The modular software architecture with varied standards and communication protocols have been successfully implemented and integrated. This model is found to be well suited to the changing requirements of the accelerator control system.

REFERENCES

- [1] Pravin Fatnani et. al. "Indus-2 Control System"- PCaPAC99, Tsukuba, Japan
- [2] PVSS-2 is a SCADA Package from ETM Austria. www.pvss.com
- [3] PROFIBUS specification manual Ed-1.0 March 1998
- [4] MatLab, The MathWorks, Natick, MA.
- [5] A.Terebilo, "Accelerator Toolbox for MATLAB", SLAC-PUB-8732, February 2001.
- [6] OS-9 Technical Manual, Microware