API MANAGER IMPLEMENTATION AND ITS USE FOR INDUS ACCELERATOR CONTROL

B. N. Merh#, R. K. Agrawal, K. Barpande, P. Fatnani, C. P. Navathe,
RRCAT, Indore, India

Abstract

The control system software needed for operation of Indus accelerators is coupled to the underlying firmware and hardware of the control system by the Application Programming Interface (API) manager. In the three-layered architecture of Indus control system, PVSS-II SCADA is being used at the layer-1(L1) for control and monitoring of various sub-systems. The layer-2(L2) consists of VME bus based system. The API manager plays a crucial role in interfacing the L1 and L2 of the control system. It has to interact with both the PVSS database and the L2. In order to access the PVSS database it uses the PVSS API, a C++ class library, whereas in order to access the L2 custom functions have been built. Several other custom functionalities have also been implemented. The paper presents the important aspects of the API manager like its implementation, its interface mechanism to the lower layer and features like configurability, reusable classes, multithreading capability etc.

INTRODUCTION

PVSS-II [1] has a highly modular structure. Various functionalities are handled by modules specifically created for different tasks. These modules are called *managers*. The Database (DB) and Event (EV) manager are the core managers that handle and manage all process variables of the system.

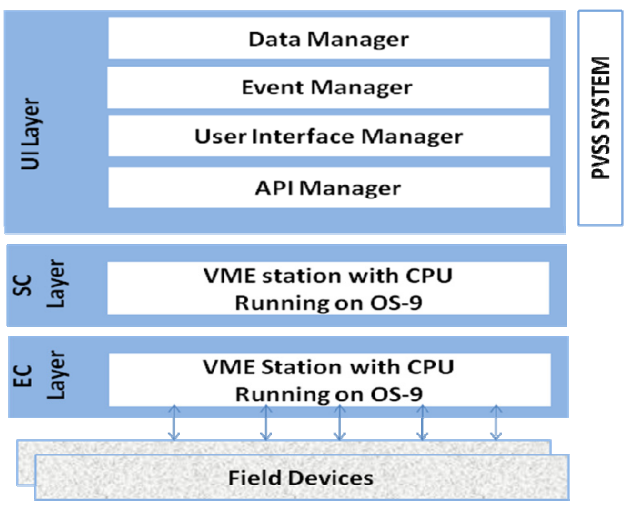

Figure 1: Software Layers of Indus-2 Control System.

#bhavna@rrcat.gov.in

The software layers of Indus-2 control system are as shown in Fig.1. PVSS system works at user Interface (UI or L1) Layer. The lower layers are Supervisory Control (SC or L2) and Equipment Controller (EC or L3) layers. The API manager is interfaced to the SC layer over Ethernet.

API MANAGER

What is API Manager?

PVSS offers a C++ application interface which enables it to extend its control functionality [2]. This interface allows the developer to implement his own custom functions together with full access to PVSS database. The self contained manager so implemented is called the API manager. It is also an interface for the integration of external programs [2]. Any software can be integrated in a PVSS System via class libraries provided by PVSS API. Implementation of these managers has been done for Indus-2 controls specific tasks. API managers have been developed for all sub-systems of Indus-2 viz. Magnet Power supply (MPS), RF, Beam Diagnostics (BDS), Timing (TCS), Vacuum (VCS), Radiation Safety (RSS), Machine safety and Interlock (MSIS), Beam line Frontend (BLFE) and Beam Orbit Correction.

Internal Structure of API

The API manager communicates with EV and DB

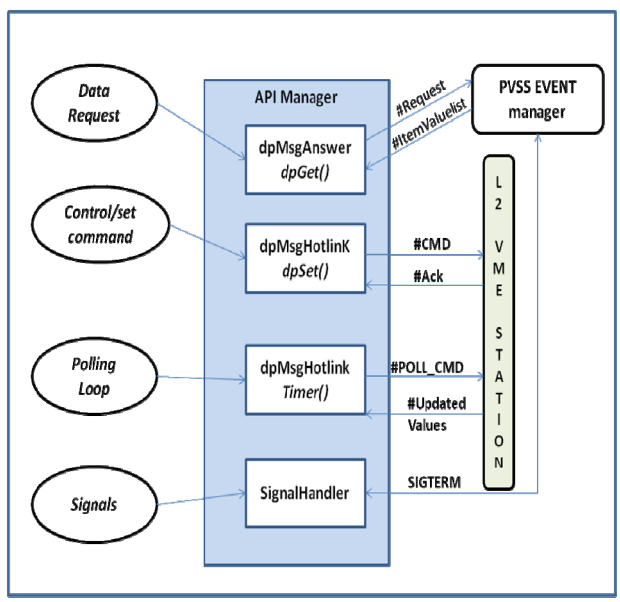

Figure 2: Event or Message handling by API.

ISBN 978-3-95450-124-3

using messages [2]. The manager processes incoming messages in the doReceive() function. doReceive() is automatically called by the *dispatch()* function for each incoming message. The *dispatch()* function is called at regular interval in the API manager.
Each message is composed of group(s) (DpMsgAnswer, DpHLGroup class) of DpVCItem (Data Point(DP) value changes) [3].
Fig. 2 shows the occurrence of events and data flow between EV, API Manager and L2.
As in Fig. 3 the DpMsgAnswer class object is returned if a request for DP values is made, whereas DpHLGroup object is returned on any spontaneous DP event.
In case of most functions regarding the access to data points, messages are handled with the aid of WaitForAnswer objects [4].
So, in order to receive notification for any DP event HotlinkWaitForAnswer is specified as:
*Static PVSSboolean dpConnect(const DpIdentifier &dpId, WaitForAnswer *wait, PVSSboolean del = PVSS_TRUE);* [4]

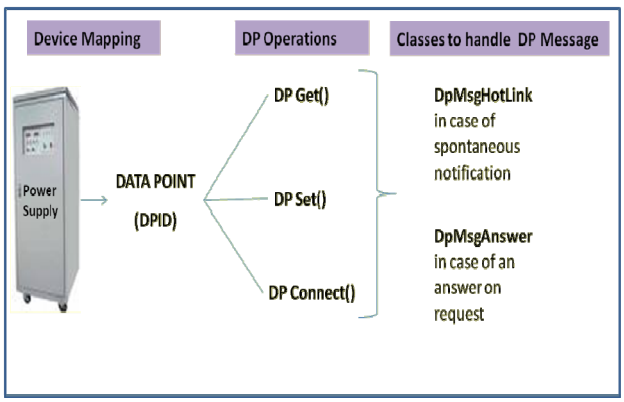

Figure 3: DP Message Handling in API.

Indus-2 API Manager Functions

When the API manager is first started it executes its initialization routines.

- ***Load device configuration information*** – Loads device related information like signal name, L3 card number, channel number and corresponding Data Point information from the configuration file.
- ***Interface to PVSS DB*** – Connect to data and event manager and for each data point get the identifier.
- ***Establish connection to L2*** - The API manager interfaces to L2 over custom TCP/IP application layer protocol.

Once initialization process is run successfully, the API manager proceeds further with its routine tasks as follows:

- ***Polling*** - It periodically polls the L2 for status and current operating values of the various devices of a sub-system. The polling is done at 1 Hz. The polling command is sent on receiving a hotlink (Fig. 2) from the global timer which is used system–wide to maintain synchronization among all the API mangers.
- ***Data filtering***- The API manger after parsing the data, does the comparison between old and new values of incoming data from L2 and sets only the changes in order to reduce the network traffic and PVSS processing.
- ***Data transformation*** – Data is first transformed from protocol specific type to basic C++ types and then to PVSS data type.
- ***Conversion*** – All incoming data values from L2 are scaled to their specified ranges.
- ***Handle control commands*** – Any device settings command or control command is received by the API manager and a command is framed according to the decided protocol. This is sent to L2 (Fig. 2) and acknowledgement received.
- ***Event log*** – All events like control commands, set commands, error received from L2, communication break from L2, operator actions from GUI are logged in the event log file in chronological order. The event log file is periodically renewed after it reaches it size limit.

Indus-2 API Manager Features

- ***Object oriented approach*** – The object oriented programming of the API allows making full use of its many benefits like maintainability, reusability, extensibility etc. The code is easy to understand and manage. In Indus control system API managers the TCP class, the timer class, the event log class all are reusable classes used system–wide for all API managers. This facilitates quick development. Fig. 4 illustrates the class diagram of API manager for magnet power supply system.

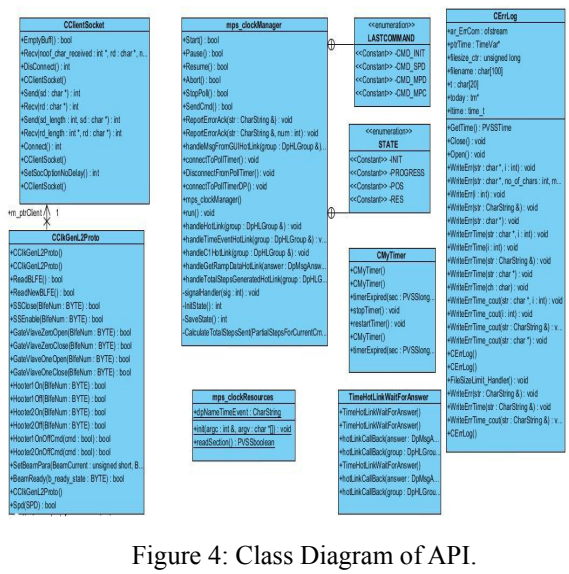

Figure 4: Class Diagram of API.

- ***Easily configurable*** - At the start-up the API manager loads information from the configuration files. These files have device related information viz. signals coming from the device, the physical

ISBN 978-3-95450-124-3

mapping of signal by card type, card number and channel number, the data point mapping of the signal by DP name. Thus any addition/modification /deletion of any signal need not require any change in API code. The manager is re-run to load a new configuration.

- ***Multi-threading to publish data*** – The API manager also serves the programs external to PVSS system with required data like beam current, beam energy, beam position from all BPIs etc. It caters to multiple clients by following a multi-threaded design.
- ***State Based***- For a special requirement like ramping the magnet power supplies, API manager code was implemented such that it retains the state of the last operation. The various states maintained are INIT, RAMP-ON, RAMP-PAUSE, RAMP-RESUME and RAMP-OVER. So even if the API Manager is re-run during ramping process, it will not affect the clock generation and ramp operation.
- ***Data Refresh on request*** – The data in the respective Data Points is usually set from the API manager only on change. All changes are reflected on the operator panel. At times it is required to refresh data which effectively will set the same data in DP but renew its timestamp. At any instance the API sets/refreshes all data on request from the operator.
- ***Error handling*** – In case of error in communication between L1 and L2, the API manager disconnects from the global timer which stops polling. It then tries to periodically establish connection to L2.
- ***Diagnostics*** – The API manager provides different diagnostic information like, the status of L3 stations, status of connection between L1 and L2, status of any control command sent from the operator interface and sequence of actions being taken.

Table 1 illustrates the DP load handled by various API manager of Indus-1 (I1) and Indus-2 (I2) control system.

Table 1: Total DP Load handled by API managers.

System Name	No. of Devices (approx.)	Total DP handled by API (approx.)
I2- MPS	174	6960
I2-RF	25	320
I2-VCS	220	1350
I2-TCS	10	145
I2-BDS	80	235
I2-RSSS	81	378
I2-BLFE	270	756
I2-MSIS	-	165
I1-MPS	97	112

CONCLUSION

The Indus-2 API managers have been developed and commissioned in 2005. Since then augmentations and new additions have been made. All features and functionalities mentioned in the paper have been implemented all through these years. API managers have been running with no crash event being reported. The load of the over all system has been nearly constant and lies between 17-21% with API manager load is maximum 2%. Possible extensions to the API are developing a generic API which caters to all future up-gradations of the Indus control system.

REFERENCES

[1] PVSS-II is a SCADA package from ETM, Austria.
[2] PVSS Driver Development by ETM.
[3] H. Milcent, P. Burkimsher, W. Salter, "How Do I Write a Driver? Ctrl Managers, API Managers and Drivers", Release 2.1, 2003.
[4] PVSS-II API Documentation by ETM.

ISBN 978-3-95450-124-3