European Coordination for Accelerator Research and Development

PUBLICATION

GUI Application for ATCA-based LLRF Carrier Board Management

Wychowaniak, Jan (TUL) *et al*

08 May 2011

The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project EuCARD, grant agreement no. 227579.

This work is part of EuCARD Work Package **10: SC RF technology for higher intensity proton accelerators and higher energy electron linacs**.

GUI Application for ATCA-based LLRF Carrier Board Management

Jan Wychowaniak, Dariusz Makowski, *Member, IEEE*, Pawel Predki, Andrzej Napieralski, *Member, IEEE*

***Abstract*–The Advanced Telecommunications Computing Architecture (ATCA) standard describes an efficient and powerful platform, implementation of which was adopted to be used as a base for control systems in high energy physics. The ATCA platform is considered to be applied for the X-ray Free Electron Laser (X-FEL), being built at Deutsches Electronen-Synchrotron (DESY) in Hamburg, Germany. The Low Level Radio Frequency (LLRF) control system is composed of a few ATCA Carrier Boards. Carrier Board hosts Intelligent Platform Management Controller (IPMC), which is developed in compliance with the PICMG specifications. IPMC is responsible for management and monitoring of sub-modules installed on Carrier Boards and pluggable Advanced Mezzanine Card (AMC) modules.**

The ATCA Shelf Manager is the main control unit of a single ATCA crate, responsible for all power and fan modules and Carrier Boards installed in ATCA shelf. The device provides a system administrator with a set of control and diagnostic capabilities regarding the crate and its sub-modules. These capabilities offered by Shelf Management are available for operators and can be further processed by higher layer applications.

This paper presents a software component, the purpose of which is to support the management and supervision processes of the ATCA crate and its sub-modules, including ATCA Carrier Board devices with AMC modules. The application allows to acquire detailed information regarding status and parameters of crucial devices (e.g. power supply voltages, temperatures, presence of reference clocks). The combination of information supplied from Shelf Manager with graphical environment and user interface of the application provides visual representation of selected system components and contributes towards efficient control and supervision activities over Carrier Board and entire ATCA-based platform.

I. Introduction

THE Advanced Telecommunications Computing Architecture (ATCA) is considered to be used as a foundation for the Low Level Radio Frequency (LLRF) control system of the X-Ray Free Electron Laser (X-FEL). The implementation of this standard will supersede the Versa Module Eurocard (VME) architecture, as this new architecture delivers exceptional reliability and availability and it is expected to win the industry support in the near future [1].

The main unit of control and monitoring of the ATCA crate and its sub-modules is the ATCA Shelf Manager (ShM) [2]. The ShM unit provides a set of interfaces for crate-external use, which enable remote supervision and control. Via those interfaces one can connect to a crate and ensure its proper operation or issue commands to its sub-modules [3]. Such possibilities are useful also at the implementation development stage, as the assumptions regarding reliability require all potential errors sources to be identified and neutralised.

Among the interfaces made available by a ShM from Pigeon Point Systems, there are such as the Web Interface and the Command Line Interface (CLI) [4]. The former is accessed via a web browser. Interaction with information obtained from a crate occurs by inspecting the contents of a web page generated by a ShM in response for a crate operator's request. In order to build a request an operator uses HTTP-based forms and buttons. Each intention to obtain data or issue a command to a crate forces an operator to manually interact with those elements, and the to examine the output from a crate in a form of response web page newly generated by a ShM.

The CLI offers a text terminal-oriented way of communication. By using this interface a crate-external workstation can become a terminal for a ShM, by which an operator can issue messages in a form of text commands with parameters, and in a similar form a ShM can respond. The interface defines a list of commands together with possible parameters, for an operator to choose from. Similarly, as it was with the Web Interface, each query or command to a crate imposes on an operator to choose an appropriate interface command, to determine parameters and then to interpret a variable amount of text constituting the ShM's response.

When it comes to monitoring of a crate operation, this need of assembling a request manually and parsing responses each time causes the supervision to be inefficient, especially when it is expected to be real-time. Therefore there is a need for a tool enabling automation the process of crate supervision and taking over the responsibility for parsing a ShM output, extracting a relevant information from it and presenting it to an operator in a clear and comprehensible manner, e.g. with using a graphical user interface. A software tool presented in this paper is designed to operate from a remote personal computer connected to a ShM, from where it enables to monitor the crate and its sub-modules and to inspect their operation conditions. The benefit from using this application is that an operator receives clear and understandable information resulting from real-time supervision of the crate inventory, which enables them to perform efficient diagnostics. This is appreciated at the stage of implementing the ATCA standard and performing the accompanying testing activities, as it can be instantly checked, whether some crate component, which is newly prepared or undergoing preparations, functions and cooperates with a crate as intended (e.g. is visible to a ShM or preforms activation/deactivation processes gracefully). This is also appreciated when an implementation is ready and operational, as remote supervision can be performed.

J. Wychowaniak, D. Makowski, P. Predki, A. Napieralski are with the Technical University of Łódź, Department of Microelectronics and Computer Science 93-590 Łódź, Poland (e-mail: jwychowa@wpk.p.lodz.pl).

II. Requirements for the Application

Direct use of the control and supervision interfaces, which are provided with the Pigeon Point Shelf Managers, is not applicable for real-time crate monitoring. Nor is it suitable for effective diagnostic activities, as it demands certain level of effort to be dedicated to formulating queries and commands to be issued to a ShM. A tool that is expected to be a solution to these issues should undertake the task to automatically formulate commands targeted at the crate parameters of interest, and to be able to issue them periodically in order to preserve having the most current values of those parameters, if a constant real-time focus upon them is needed. In addition, the tool should be able to process and present the data obtained from a crate in a visual way, with using a graphical user interface (GUI), with the aim of displaying information in coherent way. The tool should also be able to be used both with direct connection of the occupied workstation to the ShM, as well as for remote operation.

As for the demands that must be met for the tool to work, the ATCA specification states, that ShM should be equipped with a hardware link to an external entity referred to as a System Manager. This link can be used as a means of coordination of operation of multiple crates in some large system, but it is also a way through which an operator (be it a programmatic entity or a human) can connect to a ShM for control and diagnostic purposes. This System Manager link is Ethernet-based [2]. It provides a crate operator with the access to the ShM's management and supervision interfaces (CLI, Web, other). The assumption, upon which the described application is based, is that it will operate from a workstation connected via Ethernet channel to a ShM's System Manager interface, acting as a programmatic crate operator between a human one and a crate. For the purposes of inspecting a crate it will take advantage of the CLI and its diagnostic capabilities [4].

III. Details of the Application

The presented tool is being developed with the use of the 6-slot ATCA crate of the Department of Microelectronics and Computer Science (DMCS) of the Technical University of Lodz, which is equipped with a ShM based on the ShMM-500 Shelf Management module from Pigeon Point Systems.

A. Functionality and Capabilities

As mentioned, the application is in essence a supervision-oriented solution for an ATCA crate.

It can provide an operator with an insight into a crate from the ShM's view. In this perspective, it is able:

- To describe Field Replaceable Units (FRUs) visible to a ShM within a crate. For each detected FRU it reads its hot-swap state, determines an Intelligent Platform Management Interface (IPMI) address, under which a unit is visible for a ShM, and extracts from its Sensor Data Record (SDR) a set of its characteristics, such as device type (FRU types are specified in accordance with the IPMI specification) or identification string data. The acquired information is arranged by the application to a tabular form, from which an operator can learn about all the FRUs the ShM was able to identify.
- To scan for Intelligent Platform Management Controllers (IPMCs) visible to a ShM within a crate, and thus to describe all the intelligent FRUs the ShM identifies. For each such FRU it establishes its hot-swap state, determines an IPMI address, and queries the SDR for the IPMI ‘logical device’ commands and functions that the controller supports. The information is also presented by the application in a tabular form, in which an operator can view all the intelligent FRUs the ShM could detect.

The above mentioned functionality helps to determine, whether sub-modules being a part of a crate are being discovered and properly identified by its ShM, or whether they traverse the activation/deactivation stages along with the expectation. This is particularly useful during the debugging process, when it needs to be checked whether a newly assembled implementation part is operational.

The application also focuses on the state of a crate from the operator's point of view. It can reproduce the crate front panel in a form of a graphical representation and as a part of the application GUI. This visualisation indicates the presence of the hot-swappable crate components an operator can insert or replace, that are visible to a ShM. It also presents the status those components have, including their hot-swap state, or indicating the components being ready for operation or experiencing difficulties during the activation process. The graphical front panel representation works on a real-time basis. The application periodically generates and issues to a ShM queries regarding the state of selected crate sub-modules. This functionality is useful as a means of remote supervision of a working system, as well as at the stage of developing and debuging the implementation, when it enables for an immediate inspection of a crate behaviour after e.g. a newly introduced component has been inserted.

One additional piece of functionality paid by the application goes somewhat beyond the boundaries of the monitoring-oriented scheme. It is the ability to issue custom-chosen IPMI messages to an arbitrarily selected IPMC within a crate. In case of a need from an operator to acquire a specific piece of information from a given IPMC or to trigger an action, the application offers an interface enabling the operator to compose and issue appropriate messages, as well as presenting back the information the targeted IPMC responded with. As using this interface, an operator is supposed to pick an IPMC of choice and to provide an appropriate Network Function Code (NFC) and Command Code (CC) [5]. This interface enables an operator to perform control over a selected intelligent FRU, which is of significant importance, because if a targeted IPMC implements a set of custom, OEM-provided IPMI commands that can step beyond the ATCA standard, an operator is given an access to it via the interface.

This piece of the application functionality is particularly useful for Carrier Boards dedicated for LLRF [6], as those boards are often provided with additional, non ATCA-defined features, the purpose of which is to facilitate additional means of influencing their operation. Exemplary features may concern firmware update on the pluggable AMC modules as well as in the FPGA sub-circuits of a board itself, or managing and switching data transmission interfaces that a board may be equipped with. Defining and using a set of additional, board-specific IPMI commands enables for effective control in regard to such activities. The described application interface for IPMI messaging provides access to such commands, therefore enabling for management of those board-specific features from the application.

B. *Operation Principle*

The principle of operation of the application is based on a connection between the ShM and an external PC. The ShM's Ethernet interface to the System Manager is used for this purpose. The ShMM-500 ShM module, used for development and tests of the application, is equipped with a Dual 10/100 Mbit Ethernet connection giving access to several shelf-external interfaces implemented on Shelf Manager, including mentioned CLI, Web and also other, like Remote Management Control Protocol (RMCP, required by ATCA) and Simple Network Management Protocol (SNMP). The Shelf Manager operates on Monterey Linux, a specialized Linux distribution for the ShMM-500 and the ShMM-1500 Shelf Managers [3]. The Monterey Linux itself is based on a recent Linux 2.4 kernel [8].

The System Manager is defined as a logical concept capable of including software as well as human operators, and the Shelf Manager may implement the interface for the System Manager, which provides a rich set of options bringing an array of mechanisms of access to selected sorts of control and information regarding a crate.

The application, as mentioned earlier, takes advantage of the CLI. The CLI is basically a means of communication with the IPMCs across the crate as well as with the ShM itself, with the use of text-based requests and responses. The CLI is an IPMI-based set of commands that are suitable for being utilized either by a human operator or by a higher-level management application. The CLI can be accessed through Telnet or the ShM serial port. With the CLI, an operator is able to acquire information regarding the state of a crate including visible FRUs, values of sensors and their threshold settings, events, as well as general crate operation conditions.

The Monterey Linux running on the Shelf Manager hosts a vendor-provided application called Command Line Interface Agent (CLIA), which enables the CLI to be used to form and exchange the IPMI messages in a non-complicated manner [4].

The application upon launching establishes a Telnet connection to the ShM Linux via the System Manager link (Fig. 1). When connected, it interacts with the CLIA application in order to issue and receive messages provided by the CLI mechanisms.

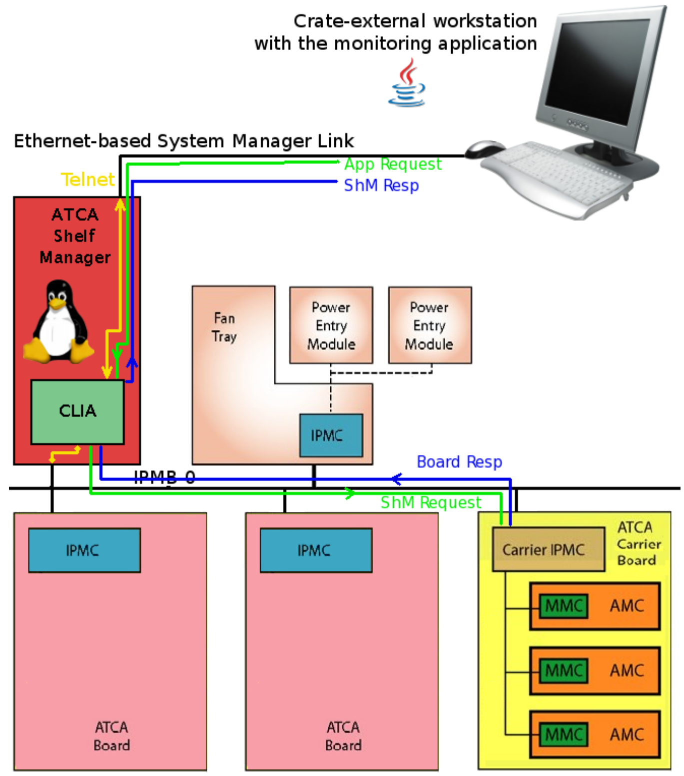

Fig. 1. Diagram of cooperation of the application with an ATCA ShM.

C. *Cooperation with Shelf Manager*

The application takes advantage of a particular set of the CLI commands. The following is a short summary of them.

- *ipmc* - The command attempts to gather and return information regarding an IPMC present under a specified IPMI address, or all IPMCs as seen by ShM, if the address is not specified. The application does not provide any addresses, as the purpose is to obtain the full list of the visible controllers. As the result of using this command the application acquires a set of entries, each corresponding to a single IPMC that was detected by ShM. Each such entry provides information, such as the IPMB address of the corresponding IPMC or the hot-swap state, as well as a group of other selected items read from SDR of the intelligent FRU that the IPMC governs. This information is extracted and processed by the application to be later presented to the user in a human-friendly form of a table.
- *fru* - The command gathers information regarding all the crate FRUs seen by the ShM. Therefore the result of using this command is similar to the aforementioned ipmc command, but does not limit to intelligent FRUs only. The data this command acquires with regard to every FRU is also similar to the set resulted in by the ipmc command. It includes the IPMB address for every FRU, its hot-swap state, and a short description of the type a particular FRU represents.
- *board* - The command offers output that is similar to the one of the fru one. However, this command also enables its user to refer to an ATCA board present in a

crate by the physical slot number that the board occupies. This way it is handy for an operator to take the slot-oriented perspective in order to determine, whether a particular slot is occupied by a board or vacant.

- *fans* - The command asks a ShM to present the information regarding presence of cooling units within a crate. ShM responds with a list of all the cooling devices it is able to detect, together with their IPMI addresses.
- *sendcmd* - Upon this command the application interface that enables to issue custom-chosen IPMI messages to an arbitrarily selected IMPC within a crate is founded. Its functionality also allows to transmit a portion of custom data if such are supplied by an operator. This data can be further processed by a target IPMC in an OEM-defined way, which may step beyond the ATCA standard, if such capabilities are implemented by the IPMC. The command receives the NFC, the CC and optionally the custom data from the corresponding application interface. Then it responds with completion code notifying of the result of the command execution and with particular information retrieved from a selected FRU sub-module, if such was requested.

An operator's interaction with the CLIA consists in typing the commands optionally accompanied with a set of parameters. The described application performs the role of a programmatic operator, which acts by automating the process of assembling and issuing those commands, and then by intercepting and interpreting their output.

D. Internal Structure

The application consists of two elements:

- A lower-level Communication Library containing networking and data exchange routines.
- A higher-level layer responsible for data processing and providing a GUI, referred to as the Data Processing and User Interaction Layer.

The former has been developed in the C++ programming language, the latter is Java-based. Although written with different programming languages, those components are capable of cooperating properly via the Java Native Interface (JNI). The JNI is a part of the Java platform, which enables applications to use native code written in C and C++.

Communication Library - This component holds the responsibility of establishing and Telnet sessions and taking care of connection-specific issues, so that the higher-level layer does not need to be aware of the networking and data transmission mechanisms. The library manages the network sockets used for communication with ShM via its System Manager interface, as well as ensures that data are transmitted in form suitable for the Telnet protocol. Telnet is generally a means for text-oriented data exchange, the aim of which is to provide a way for a workstation to become a terminal for another, remote one, thus to become its input and output device. This text-oriented communication imposes certain rules regarding treating of special characters. Such a character is the one, the task of which is to simulate pressing enter on a remote host. The communication library encapsulates all the issues of this kind, which allows the higher-level layer to be completely separated from the communication aspects, and fully focused upon the ATCA crate domain.

Data Processing and User Interaction Layer - The activities of this component facilitate the actual crate-focused functionality. This includes formulating queries regarding particular crate aspects. The queries are sent to the communication library in order to be transmitted to a ShM. When the crate response is returned by the library, it undergoes processing in order to extract the information of interest. This information is then prepared to be graphically presented.

The component is divided into three sections referred to as a processing block, a data composition block and a GUI block (Fig. 2). The first one, the purpose of which is to scan the incoming text portions from a ShM in a search for relevant pieces of information, is based on regular expressions. It incorporates a set of patterns, each one corresponding to a single CLI command the application uses. Basing on a particular pattern the block is able to determine which pieces of information from the response to the command, that the pattern is dedicated to, is relevant. The information extracted from a single or multiple responses is then put together into a data structure by the data composition block, and sent to the GUI block for presentation.

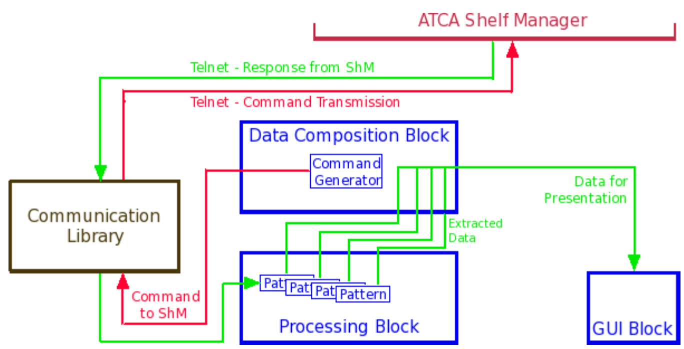

Fig. 2. Dataflow between the application components.

The cooperation of the three component blocks can be described basing on a single query cycle, a purpose of which is to obtain data required to update the crate front panel representation that is being graphically displayed to an operator. The data composition block is also responsible for formulating the commands to be sent to CLI on ShM, and manages information required for this process. The application assigns a dedicated thread to a task of keeping the front panel representation up-to-date. This thread periodically triggers a data composition block's activity of generating a set of commands, the responses of which are to be used for refreshing the front panel representation. The commands are then sequentially sent for transmission, with collecting a response after each one. The incoming responses are interpreted by the processing block and the obtained information is packed by the data composition block to a form, in which it is expected by the GUI block. First of the set of commands used for each front panel representation update cycle iteration is based on the board CLI command. Its purpose is to consecutively scan board slots of a crate to determine which are occupied by Carrier Boards. For all the occupied slots a command based on fru is issued in order to acquire further information regarding state of a Carrier, as well as the presence and state of AMC modules the Carrier may be equipped with. When done, commands based on fans and ipmc are sent to investigate the cooling devices. All the information from responses to this series of commands, after having been prepared by the processing block and the data composition block, are passed to the GUI block for presentation (Fig. 2).

IV. CONCLUSION

The ATCA standard is a suitable candidate for being the successor to VME and also an appropriate choice for a foundation of the control system for high-energy physics experiments like X-FEL. As the ATCA-based system is complex, but at the same time it needs to deliver an extremely high reliability level, the efficient monitoring of its elements is vital. The crate supervision tools currently offered by the vendor-provided ShM firmware are not satisfactory for this purpose, and in particular, not applicable for operation in real-time. The solution being the subject of this paper addresses that problem by adding the missing real-time diagnostic capabilities and providing faster and more efficient access to key information regarding status and operation of an ATCA crate and its sub-components. It also facilitates possibilities for performing custom management of Carrier Boards for LLRF.

The development and tests of the presented application are being conducted with the use of Pigeon Point's ShMM-500-based Shelf Manager, together with ATCA Carrier Boards from RadiSys and custom Carrier Blades by DMCS. The initial acceptance tests are proving that the application is capable of successfully fitting its purpose, which means the efficiency of operator's supervision-related activities over the ShM's external communication interfaces is improved. It is expected that the tool will be helpful during the process of creating custom ATCA components, as well as later, at the stage of monitoring of the operational system. Therefore it its safe to make an assumption that it will contribute to the reliability level that the ATCA standard implementation for X-FEL is expected to deliver.

For the time being, the application is focused upon supervision and diagnostics, with a rather basic control means in a form of facility for generating arbitrary IPMI requests. Further development plans are intended to focus upon upon extending its management capabilities, which will allow to remotely affect conditions and modify parameters of the operation of a crate and its selected sub-modules.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Commission under the EuCARD FP7 Research Infrastructures grant agreement no. 227579. The author is a scholarship holder of project entitled "Innovative education ..." supported by European Social Fund.

REFERENCES

[1] Jezynski, T.; Simrock, S. 'The low level radio frequency system architecture for the european X-FEL.' Proceedings of the International Conference on Mixed Design of Integrated Circuits and Systems, 2006. MIXDES 2006.

[2] AdvancedTCA PICMG 3.0 Revision 2.0, “AdvancedTCA Base Specification”, 28 October 2005

[3] Pigeon Point Systems 'Shelf Manager User Guide', Release 2.6.1, December 1 2008. [Online]. Available: www.pigeonpoint.com

[4] Pigeon Point Systems 'Shelf Manager External Interface Reference' Release 2.6.1 December 2, 2008. [Online]. Available: www.pigeonpoint.com

[5] Intel Hewlett-Packard NEC DELL, “Intelligent Platform Management Specification version 1.5”, Document Revision 1.1, 20 February 2002

[6] Zawada, A., Makowski, D., Jezynski, T., Simrock, S., Napieralski, A., "ATCA Carrier Board with IPMI supervisory circuit", Mixed Design of Integrated Circuits and Systems, 2008. MIXDES 2008. 15th International Conference, pp. 101 – 105, June 2008

[7] B. Eckel, "Thinking in Java", 4th Edition, Prentice Hall, 2006

[8] Pigeon Point Systems 'Monterey Linux User’s Guide' Au1550 Edition, Release 2.1.2. [Online]. Available: www.pigeonpoint.com

[9] Zawada, A.; Makowski, D.; Jezynski, T.; Simrock, S.; Napieralski, A. "Prototype AdvancedTCA Carrier Board with three AMC bays", Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE, pp. 53 – 57, Oct. 2008

[10] Makowski, D., Koprek, W., Jezynski, T., Piotrowski, A., Jablonski, G. Jalmuzna, W., Simrock, S. 'Interfaces and communication protocols in ATCA-based LLRF control systems.' Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE, pp. 32 – 37, Oct. 2008

[11] Przygoda, K.; Piotrowski, A.; Jablonski, G.; Makowski, D.; Pozniak, T.; Napieralski, A. 'ATCA-based control system for compensation of superconducting cavities detuning using piezoelectric actuators' Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE, pp. 38 – 43, Oct. 2008