IMPEMENTATION OF DATA ACQUISITION SYSTEM FOR OPERATING CONDITION IN THE 100 MeV PROTON LINAC*

J. H. Kim[†], Y. G. Song, H. J. Kwon, Y. S. Cho, KAERI, Gyeongju, Korea

Abstract

The goal of data acquisition system is to provide an efficient user interface to analyze operating condition of the KOMAC linac. The KOMAC control system which is based on Experimental physics and Industrial Control System (EPICS) has been monitoring and archiving the operating condition using Channel Access (CA) protocol [1]. A data acquisition system has been developed with Qt framework that accesses EPICS IOCS and MySQL database via EPICS CA protocol. The data acquisition system provides calibrated data and efficient function to analyze data easily. In this paper, we are describing the implementation of data acquisition system for operating condition in the 100-MeV proton linac.

INTRODUCTION

The Korea Multi-purpose Accelerator (KOMAC) has been operating the 100 MeV proton linear accelerator. The KOMAC control system, which is a monitoring and control for the linac and beam lines, is implemented as a soft real-time distributed control system based on Experimental Physics and Industrial Control System (EPICS). EPICS is a set of Open Source software tools, libraries and applications developed collaboratively and used worldwide to create distributed soft real-time control systems for scientific instruments The EPICS-based control system consists of several subsystems such as timing system, cooling system, alarm monitoring system and etc. The structure of KOMAC control system is illustrated in Fig. 1.
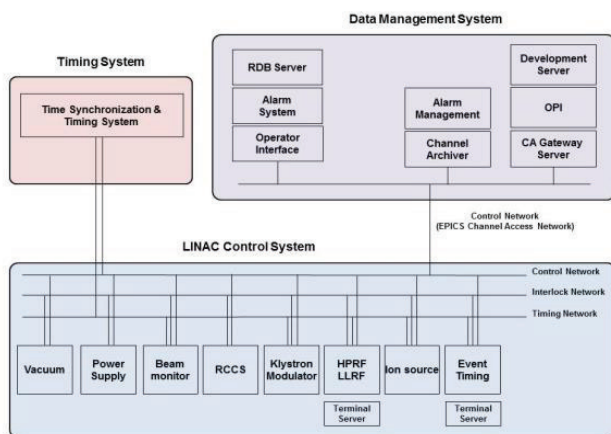

Figure 1: The structure of KOMAC control system.

As EPICS is used by many large scientific facilities throughout the world, many supports and applications have been developed. KOMAC chooses a Control System Studio (CSS) that is an Eclipse-based collection of tools to monitor and operate large scale control systems

with EPICS [2]. A user interface for KOMAC was configured with CSS primarily. As The 100 MeV linac has been upgraded, the user interfaces that are easily used and user-friendly have been issued. For user's convenience, data acquisition applications have been developed with EPICS toolkits and an alarm monitoring system are in test. The alarm system is configured with eclipse, JCA, CAJ and Maria-connector [3].

ALARM SYSTEM

An alarm system monitors conditions and malfunctions of software and hardware and announces problems to operator with sounds and display before fatal effect on linac occurs. The KOMAC alarm system is configured with Best Ever Alarm System Toolkit (BEAST) using CSS application [4]. The KOMAC alarm system is illustrated in Fig. 2.
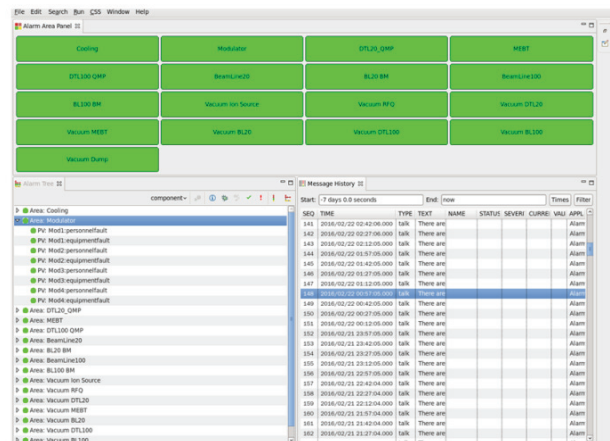

Figure 2: The KOMAC alarm system.

The alarm system takes field information such as LOLO, Low, HI, HIHI of Process Variables (PVs) from EPICS Input Output Controllers (IOC) and visualizes the status with colour in alarm area panel. We defined status with color: a green light means normal and Red is major alarm. The alarm area panel display the current alarm state of top-level of the alarm tree hierarchies. The hierarchical structure of alarm system is displayed on the alarm tree. The alarm tree consists of area, system and PV. The Areas are top-level elements, such as modulator, vacuum of the alarm hierarchy and below the area, PVs are arranged. Operators check the alarm trigger PVs generate actual alarms and add, rename and remove alarms. The currently active alarms are checked using alarm table. Based on the BEAST alarm system, an alarm system has been implemented to operate 100 MeV linac more efficiently. The new alarm system for KOMAC is illustrated in Fig. 3.

Figure 2: The New alarm system for KOMAC

The alarm system is implemented using jdk1.8 in eclipse mars, JCA library, CAJ library and Maria DB. The alarm system also displays alarm status with colors and white is added, means acknowledge. The configuration of the alarm system is illustrated in Fig. 3.
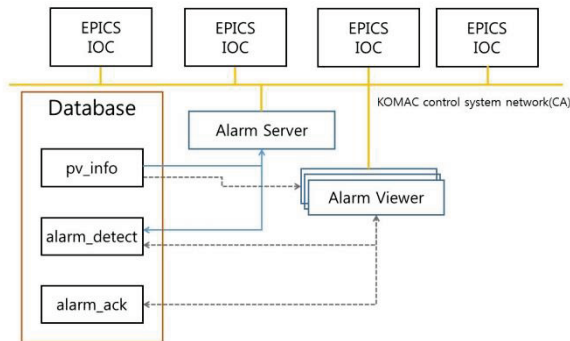

Figure 3: The configuration of the alarm system.

Database Schema

The Alarm system database is based on mariaDB. Java can access DB using mariadb-connector library. The database schema consists of pv_info tables, alarm_detect tables, alarm_ack tables and group_info tables. The specification is shown in Table 1.

Table 1: Database Schema Specification

Pv_info	Alarm_detect	Alarm_ack
id	id	id
pv_name	a_date	al_date
alarm_type	status	ack_date
user_major_val	value	ack_user
user_minor_val	ack	
g_id		
p_desc		
p_level		

The primary table is the pv_info table, which contains information of PVs that is monitored in alarm system. The alarm_type field concludes using PV alarm related field or alarm information, user set through alarm server program to display alarm status on alarm viewer. The alarm_level field concludes whether auto reset alarm or not. Operation modes that adjust the alarms displayed depending on the alarm_level field will be added.

Alarm Server

The alarm server is configured with alarm server based on EPICS IOC. The configuration of the alarm server is illustrated in Fig. 4.
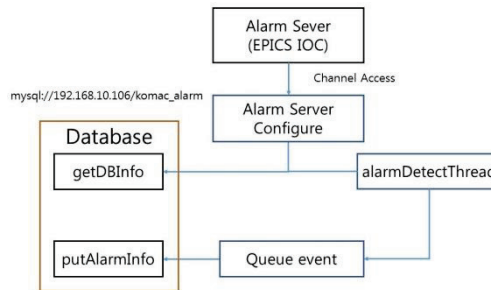

Figure 4: The configuration of the alarm server.

The alarm server takes data from pv_info table and monitors and compares PV status with data registered in pv_info table. When any event occurs, the event data, such as timestamp, severity and PV name are stored in alar_detect table. As several records are made in alarm sever, monitored PVs are reprocessed without changing original PVs. And using calc record and EPICS moudles, the alarm system combines several PV and creates a new alarm.

Archive Viewer

Alarm Viewer show alarm status of PVs that is registered in Database on alarm info panel. The alarm viewer takes event status from alarm_dectect table and displays event status. The configuration of the alarm Viewer is illustrated in Fig. 5.
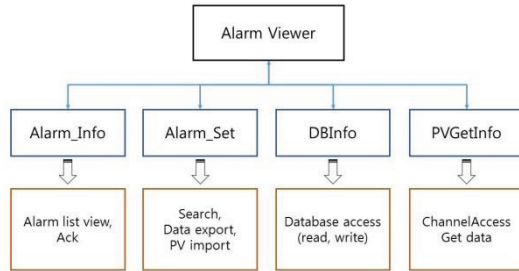

Figure 5: The configuration of the alarm Viewer.

The latest event is shown on the top and previous events are pushed down not disappeared. Operators change alarm status with clicking alarm status in alarm view and ACK button on the top. Alarm status turn white and data are archived in alarm_ack. Then alarm that are dealt with are disappeared with Clear Ack button. The search panel displays alarm history and retrieve history you want to see from database. But Alarm history and search function have not been implemented yet. Following Figure 5 shows the configuration of alarm viewer.

CONCLUSION

To provide more convenient user interfaces, high level applications have been being developed for KOMAC control system. The alarm system is implemented and the performance of the alarm system is under test, compared with existing alarm system. The alarm server program puts PV data into database in database schema. The alarm viewer display event status, taken from database.

In the future, History, search and calculating function that combines several PVs will be implemented in the alarm system.

ACKNOWLEDGEMENT

REFERENCES

[1] EPICS, http://www.aps.anl.gov/epics/

[2] CSS, http://controlsystemstudio.org/

[3] MariaDB, http://mariadb.org/

[4] Young-Gi Song*, Jae-Ha Kim, Tae-Sung Ahn, Hyeok-Jung Kwon, Yong-Sub Cho, "Improvement of Control System Infrastructure and High Level Application for KOMAC LINAC", Transactions of the Korean Nuclear Society Autumn Meeting Gyeongju, Korea, October 29-30, 2015