STUDY OF PC-BASED CONSOLE FOR THE J-PARC CONTROL SYSTEM

Norihiko Kamikubota*, Noboru Yamamoto, Guobao Shen, Kazuro Furukawa, Hidetoshi Nakagawa,
Junsei Chiba, Tadahiko Katoh, Susumu Yoshida[A)] and Makoto Takagi[A)]
High Energy Accelerator Research Organization (KEK)
[A)]Kanto Information Service Co. Ltd. (KIS)

Abstract

The control system for the J-PARC accelerator complex has been under development using the EPICS toolkit. Though we need X-window consoles with the EPICS-based control system, the only choice is to use the PC platform, because no X-terminals are presently available on commercial markets. In addition, we need Java consoles, since we expect more use of Java in future commissioning studies.

We used PCs with two operating systems (Linux and Windows), in beam-commissioning studies of the proton linac during 2002-2004. Based on this experience, we discuss the realistic perspective of a PC-based console system suitable for the J-PARC project.

J-PARC PROJECT

Overview and Status

A high-intensity proton accelerator complex, J-PARC[1], is now under construction. The J-PARC project consists of three accelerator facilities : a) a 400-MeV proton linac, b) a 3-GeV RCS (rapid-cycling synchrotron), and c) a 50-GeV MR (main synchrotron); there are also two experimental facilities : d) the Material and Life Science Facility at the RCS, and e) the Nuclear and Particle Physics Facility at the MR.

J-PARC is a joint project between two national research institutes : JAERI (Tokai, Ibaraki) and KEK (Tsukuba, Ibaraki). The construction site is Tokai. However, the pre-injector part of the proton linac was constructed at the Tsukuba site. It is comprised of a) an ion source, b) a low-energy beam transport (LEBT), c) a 3-MeV RFQ Linac, d) a medium-energy beam transport (MEBT), and e) a 50-MeV drift-tube linac (DTL). Commissioning studies with proton beams have already been carried out [4, 5]. Studies at KEK were finished, and the pre-injector part will be transported and re-installed at the Tokai site in 2005.

Development of the Control System

EPICS (Experimental Physics and Industrial Control System) is a software toolkit to make distributed real-time control systems for large accelerators. Up to now, many large accelerator institutes, including the KEKB rings at KEK [6], have introduced the EPICS toolkit [7, 8]. We also decided to use EPICS to control the J-PARC accelerators [9].

A prototype control system was developed with the EPICS toolkit. The prototype control system consists of: (a) 4-6 VME-bus computers, (b) a Linux server and a HP-UX server as the main development servers, (c) 2 BSD-based servers for supplemental functions of the main servers, (d) 2 NAS (network attached storage, each 180GB) systems, and (e) 4-10 personal computers (hereafter PC) for the console system. The prototype system was used for beam-commissioning studies of the pre-injector proton linac at KEK during 2002-2004. More detailed descriptions are given elsewhere [10, 11].

CONSOLE SYSTEM

Introduction

Since many of the EPICS standard tools have been developed with the X-window environment, it is natural to use X-terminals for the console system. However, the major commercial companies have already stopped their support of X-terminals. Instead, PCs have been widely used in various fields, including accelerator controls. Thus, we decided to evaluate PCs as our console system.

We expect a wide variety of console applications in future commissioning studies of all J-PARC accelerators. In Table 1, the expected console applications are summarized. Table 1 shows that we must prepare a Java environment for our console system in addition to the X-window environment.

Table 1: Expected console applications

Group	Tools	Environment
a) Basic EPICS tools	MEDM VDCT	X-window Java
b) Dedicated applications	Python Java	X-window Java
c) High-level applications	SAD script XAL	X-window Java

Feasibility studies of a PC-based console system were carried out with the prototype control system [10, 11]. During a commissioning in 2002-2004, both Linux PCs and Windows PCs were used and evaluated as consoles with real accelerator signals of a pre-injector proton linac.

* E-mail: norihiko.kamikubota@kek.jp

[1] Japan Proton Accelerator Research Complex [1, 2, 3]

Linux Console

Features

Using Linux PCs as consoles is natural for EPICS-based control systems. The X-window environment is available as a default feature. Java installation requires manual actions, but it works. Major EPICS tools (MEDM, VDCT, Channel Archiver, etc.) run stand-alone with Linux PCs. Multiple displays with one PC are possible with a video card by Matrox company. The sharing of files and the sharing of user accounts are standard features of Linux by using NFS and NIS, respectively.

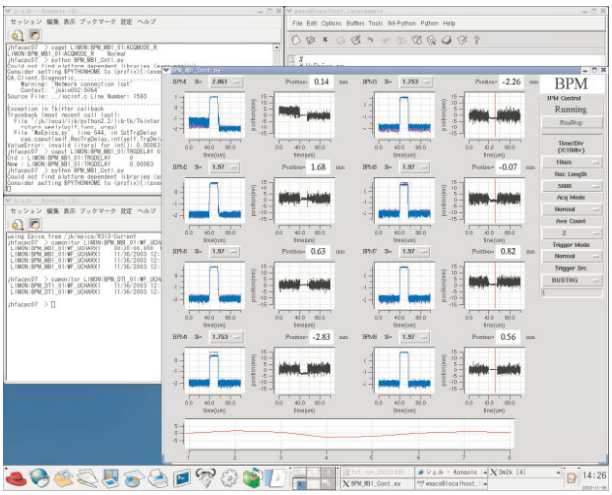

Figure 1: Screenshot of a Linux console.

Experience

We have used 4-8 standard desktop PCs as Linux consoles for beam-commissioning studies. Fig. 1 is a typical screenshot of a Linux console with a Python application (X-window based). Monitoring waveforms of beam-position monitors were demonstrated at a refresh rate of 5 Hz, which implies that Linux PCs show sufficient performance for our studies.

During a 3-year commissioning period, we experienced version-ups of Red Hat distributions,[2] which made us confused. To keep using an old version made problems when we bought new PCs, since they included new hardware that was not supported by the old version. In addition, some of the PCs caused hardware problems after successful use for 3-4 years. This fact indicates that the reliability of PC hardware is not high.

Windows Console

Features

The advantage of a Windows PC is that there is a wide variety of commercial and free software. Java is free and easy to install. The X-window environment is not the default, but there is free X-server software (Cygwin/X [12]), as well as commercial products. If necessary, the EPICS tool MEDM can run stand-alone on a Windows PC by using commercial X-server software (Exceed). Multiple display is a default feature. File sharing between Windows and Linux PCs is possible by using open-source software (Samba [13]). Sharing of the NIS-style user accounts is also possible with recent Samba versions.

In general, Windows PCs have higher risks of virus than Linux PCs. Thus, it is a problem that each Windows PC is requested to open some halls on the firewall to enable the EPICS communication protocol.

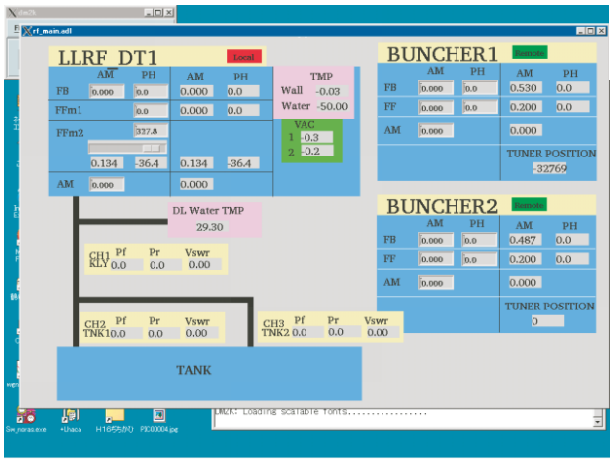

Figure 2: Screenshot of a Windows console.

Experience

Fig. 2 is a screenshot of a Windows98 console with a MEDM application (X-window based). The MEDM application ran on a Linux PC, and was displayed on the Windows98 PC by using a Cygwin/X. A waveform display at a few Hz was also demonstrated, which was slightly slower than Linux consoles. File sharing with Linux PCs was demonstrated well by using Samba 3.0.7. Sharing of the NIS user accounts with Linux PCs was realized by using NISgina[3] [14].

Though we, the control group, did not want to introduce Windows PCs into the control system, commissioning team members hoped to use Windows applications, especially Office Suite, for commissioning studies. During the beam-commissioning period, they introduced Cygwin/X into their personal notebook PCs by themselves, and displayed MEDM and Python applications on the PCs. In order to enable immediate reaction to virus problems, we prepared a dedicated network of the Windows PCs, which is separated from the main control network; the network traffic produced by the Windows PCs has been continuously monitored.

[2] We started with 7.2 in 2002, soon 7.3, a year later moved to 9.0.

[3] We started with Samba 2.2, which needed NISgina. Now NISgina is not necessary by using the PDC feature of the Samba 3.

Study of Java

Features

Java is an object-oriented language. In general, the maintenance of large-scale applications is easier than other languages. Recently, strong interest has arisen to use Java in accelerator controls. Actually, some control tools, such as VDCT and XAL, have been developed by Java, and are already being used in multiple accelerator institutes. In addition, from the viewpoint of a console system, Java has a unique feature: once developed, it runs both on Linux and Windows consoles.

Experience

When we started studies of Java for console applications, we searched for a Java-based package suitable for accelerator controls. Abeans, a Java-beans package developed by COSYLAB [15], was successfully used in the RI-Beam Factory control system at RIKEN [16]. Though the control system at RIKEN is not based on EPICS, we decided to study the feasibility of Abeans with our EPICS-based control system.

A free package, Abeans R3.0, which includes the EPICS communication protocol, was downloaded and installed in a WindowsXP PC. Free ADEs[4] (Eclipse and NetBeans) were successfully used during the development of console applications. Java distribution JDK1.4.2 was used throughout the studies. Fig. 3 is a screenshot of a Java application console with Abeans components. The beam-currents and waveforms of beam-monitors of the proton linac were demonstrated. The beam-currents were displayed at a 5 Hz refresh rate by using the pre-defined *gauger* component. However, waveforms refresh rate was as slow as a few Hz by using the *profile* component. The observed performances of the Abeans components were acceptable for us. We tried to customize the screen, especially the profile components and the trend-graph in Fig. 3; we found that it was not easy. We concluded that more different pieces of components, and more flexible customizations are needed in the Abeans components to develop realistic console applications.

[4] Application Development Environments

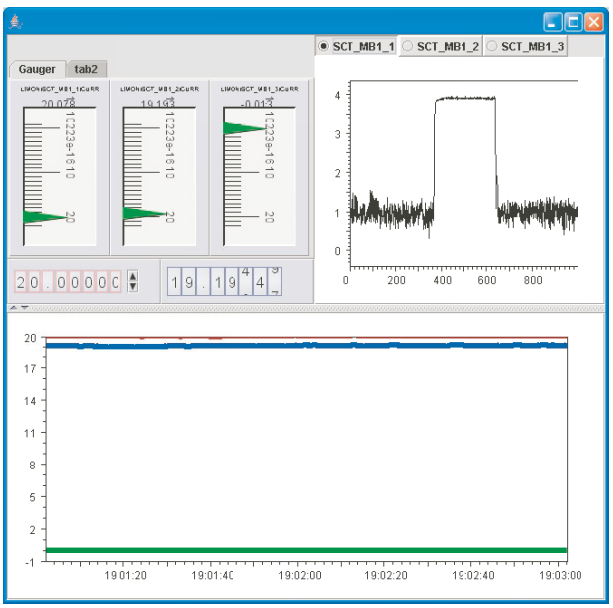

Figure 3: Abeans application.

CONSOLUSION

The console system for the J-PARC accelerators is requested to have both X-window and Java environments. We have experienced using both Linux and Windows PCs as consoles, during beam-commissioning studies of the pre-injector part of the proton linac.

Linux PCs have both X-window and Java environments, and showed sufficient performance. It seems to be apparent that using Linux PCs as consoles is a good solution for us. However, experiences have showed that we must use reliable hardware for console PCs. Industrial PCs, or thin-client PCs, would be possible candidates.

There was a strong request from the commissioning team members that Windows PCs would be necessary for beam studies. We have tested Windows PCs with a free X-server (Cygwin/X). It showed good performance for most of our X-based applications, but the waveform display was slightly slower than that of Linux PCs. Potential problems involving virus are serious with Windows PCs. We prepared a separated network only for Windows PCs, but more ideas are needed.

The availability of a Java-based application was demonstrated with a Windows PC. Free Java packages (Abeans, Eclipse, and NetBeans) were successfully used during the development of console applications. Though we satisfied performances of Java-beans components, we concluded that more functionalities are desirable to develop realistic console applications.

ACKNOWLEDGMENT

The authors would like to thank the JAERI-side control members (H. Yoshikawa, H. Sakaki and Y. Kato) for various discussions. We also thank the KEKB control staffs for their advise and suggestions. The considerable contribution on our EPICS environment comes from a member of the EPICS traversal group at KEK, J. Odagiri. We also thank T. Manaka, University of Tsukuba, for valuable discusion on Samba and NISgina. Part of the financial support of Abeans studies was provided by the fund of Joint Development Research at High Energy Accelerator Research Organization (KEK), for the joint research between RIKEN and KEK.

REFERENCES

[1] Y.Yamazaki, "The JAERI-KEK Joint Project for the High-Intensity Proton Accelerator, J-PARC", Proc. PAC2003, May 2003, Portland, p.288-290

[2] Y.Yamazaki, ”Accelerator Technical Design Report for JPARC”, JAERI-Tech 2003-044, and/or KEK Report 2002-13
[3] http://j-parc.jp
[4] M.Ikegami, et.al, ”Beam Commissioning of the J-PARC Linac Medium Energy Beam Transport at KEK”, Proc. PAC2003, May 2003, Portland, p.1509-1511
[5] F.Naito, et.al, ”Results of High-Power Conditioning and the First Beam Acceleration of the DTL-1 for J-PARC”, Proc. LINAC2004, Luebeck, Germany, Aug.2004
[6] ”KEKB Accelerator Control System”, Nucl. Instr. Meth. A499(2001)138-166
[7] R.Lange, et.al, ”EPICS: Recent Developments and Future Perspectives”, Proc. ICALEPCS2003, Gyeongju, Oct.2003, p.278-281
[8] http://www.aps.anl.gov/epics/index.php
[9] T.Katoh et.al., ”Present Status of the J-PARC Control System”, Proc. ICALEPCS2003, Gyeongju, Oct.2003, p.1-5
[10] N.Kamikubota et.al., ”Prototype Control System of the 60-Mev Proton Linac for the J-PARC Project”, Proc. ICALEPCS2003, Gyeongju, Oct.2003, p.81-83
[11] N.Kamikubota et.al., ”Experience of Prototype Control System for the J-PARC Proton Linac”, to be submitted to PAC2005, Knoxville, May 2005
[12] http://x.cygwin.com or http://cygwin.com
[13] http://www.samba.org
[14] http://nisgina.daekin.edu.au
[15] http://abeans.cosylab.com/
[16] T.Tanabe et.al., ”Current Status of the Control System Development at RIKEN RI-Beam Factory”, Proc. ICALEPCS2003, Gyeongju, Oct.2003, p.597-599