COACK-II PROJECT ON ACCELERATOR CONTROL KERNEL DEVELOPMENT

Isamu ABE, Junichi Kisiro, Kazuyuki Nigorikawa, Takashi Kosuge, Shin-ichi Kurokawa, **Masakatsu Mutoh, and **Masaharu Kitamura
High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan
*Mitsubishi Electric System & Service Co., Ltd.
**Tohoku Univ. Sendai, Miyagi, Japan

Abstract

A PC (personal computer) system[1] has been applied partially in each of the top (GUI), middle and lower (I/O, device) layers of the accelerator control system. In fiscal year 1999, further reinforcement and enhancement of the device controller PCs of the device layer was scheduled. With the advancement of existing PC architecture, as well as hardware and software, research into an accelerator kernel that has wide applications has been undertaken. It will be possible to apply this kernel to a wide variety of accelerators. The development of COACK-II[2] and the realization of this research is presented in this paper.

INTRODUCTION

For analysis purposes, the requirements for all types of accelerator control systems have been standardized in this study. The adaptable product is packaged using COM (Component Object Model) as a component-ware, and is characterized by its distributed operation (DCOM). A PC control system is achieved through the integration of SCADA[3]. Accelerator operation has been enhanced through the use of COACK (Component-ware Oriented Accelerator Control Kernel), which is the main subject of this paper. In fiscal 1999, Japan Science & Technology (JST) are supporting COACK-II development as an intensive research program.

1 BACKGROUND OF COACK-II

Research on the PC management of the Linac (Linear Accelerator) operator's console at the High Energy Accelerator Research Organization (KEK) has advanced since 1987. Our project started with financial aid from JST, to develop a practical common or standard accelerator control kernel on Windows NT using COM technology. This new kernel is based on the conventional concept of the KEK PF Linac control PC system which was developed using procedural programming, and can be applied to any accelerator or to industrial fields for measurement and control. Comparable systems have been developed at accelerator facilities in a number of countries, where the same problems have been encountered. In an analysis of the problem we experienced led to the observation of a common control kernel in various measurement and control systems, especially in the accelerator field. During our analysis, the common accelerator control kernel was a subject of controversy. In addition to the recent appearance of a new operating system, Windows NT, and the remarkable progress in the cost reduction of PC hardware, outstanding forms of technology have become available and are enjoying practical applications in various industrial sectors. Taking advantage of these new frontiers in technology enabled a next-generation control system to be devised. A control system entirely different from the type used in the UNIX era. The new features of the measurement and control systems can also find application outside of the accelerator domain.

2 ACCELERATOR CONTROL KERNEL (COACK-II)

Even though the functional requirements for all accelerators are virtually the same, an individual program for each has been developed. Trying to clearly understand the huge flowcharts and 'spaghetti-like' programs developed with procedural languages is overwhelming. Since it demands a great deal of time and effort to prepare programs, we felt that it might be better to pool this kind of software.

The first goal of COACK-II was to standardize the common accelerator functions, especially for the middle layer, by increasing the flexibility. The second was to create new characteristics, making use of the latest technology through the introduction of component-ware. The characteristics are obtained not only by defining the standard functions, but also by using components that allow experimentation on encapsulated objects. Standardization enables improved functions through flexible support. For COACK-II, an accelerator control kernel was developed (fig.1) from the viewpoint of a broad range of applications and the creation of components.

3 FEATURES OF THE COACK-II PROJECT

What new characteristics were created by adopting measuring and control functions that feature innovative forms of technology?

(1) Many discussions have centered on making a universally adaptable control kernel for accelerators, measurement and the control systems. At present, there are many types of commercially available software packages and jointly developed control system software available. But this marks the first attempt at making accelerator control kernel components in addition to standardizing common functions on various accelerators, and it is expected to set a new course in the realm of this software. This method differs completely from conventional software supply. Setting up properties and methods, that are unaffected by source level is extremely easy. By basing preparations on COM standards, it has become possible for anyone to supply widely usable components using Visual Basic 6.0.

(2) This adaptable kernel for accelerator control based on fully distributed objects, may be the world's first, developed as a form of installation technology. A customized kernel can be created, simply by downloading components from a server. Much in the same manner as collecting information from a homepage, an original system can be built by extracting usable components from the server. Component-ware dynamically operated by shifting from one place to another on a network with a fully complete binary object level was installed and demonstrated.

(3) Push technology [4] to manage assorted versions of client components and applications has been developed.

(4) A way to establish components at the GUI (Graphical User Interface) level has been developed and is being promoted in Europe. The concept has been highly evaluated. Its operation has become more flexible through the incorporation of ACOP [5] (Accelerator Component Oriented Programming), and newly integrated acceleration control has become available through a link with COACK-II. In this manner, a viable software sharing system among research institutes is evolving.

(5) Commercially available SCADA (Supervisory Control and Data Acquisition) software is rapidly improving, especially with ActiveX connections and is available at reduced cost. We can choose among them and easily build a control system. In each GUI or I/O (Input/Output) layer, accelerator control can be realized simply by linking ActiveX with SCADA. Intelligent I/O control and the GUI using SCADA can also be readily integrated with the COACK-II system.

4 SYSTEM COMPONENTS

In this COACK-II project, we focused on the middle and upper layers. From a structural point of view, the upper and lower layers are equal in the COACK-II. GUI and I/O clients are managed in the same way in COACK-II.

4.1 Functions of COACK

COACK-II main functions are:
- Client communication, message exchanger,
- Command control, Buffer, logging,
- Database (Dynamic DB, Static DB, Archiving) management,
- Integrated management of accelerator operation,
- Download server for component-ware,
- Component-ware dynamic operaion,
- Extraction and reuse of macro-operation command,
- Command sequencer,
- I/O control, Alarm system by DCOM,
- SCADA management, ACOP attachment,
- Download server for component-ware,
- Application pushing. [4]

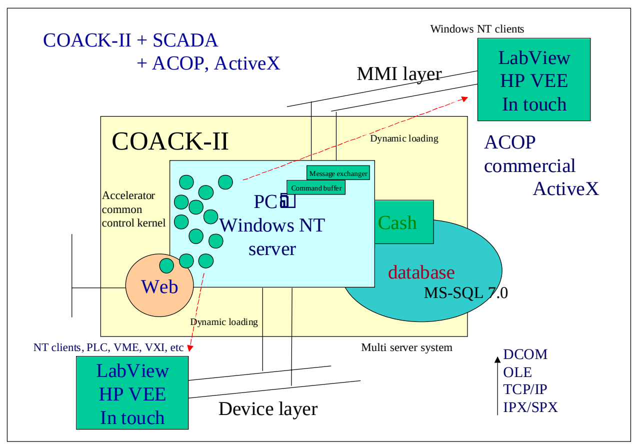

Fig.1 COACK-II system

4.2 SCADA

A Basic idea of the COACK-II system is to apply as much SCADA software in the upper and lower layers as possible. One might say that SCADA and COACK-II were meant for one another. COACK-II makes Integrated SCADA operation possible either as a larger scale control system or as an intelligent control system.

4.3 Upper layer

In the connecting pattern of MMI, the upper layer is easy to link using ActiveX, and various kinds of commercial software are available on the market. Conceivably, the introduction of many kinds of components is also possible. Apart from ActiveX or DCOM (Distributed Component Object Model) communication, that is; clients who do not utilize Windows, COACK-II provides socket communication services. Further, for clients who are not COM users, COACK-II also permits the application of almost equivalent functions. Assuming that Windows NT is used, a wide variety of Windows components can be loaded. Also, COACK-II makes it possible to introduce not only a VB (Visual Basic) container, but also SCADA software packages, such as LabView, VEE, Intouch, etc. as components.

4.4 ACOP connection

In Europe a particular component for accelerators has been developed at DESY/CERN. The concept has been highly acclaimed in the PC domain. It can be simply downloaded from a server and with configuration, implemented on any accelerator. Experimental test results between ACOP, LabView and VB container have indicated its stability and ease of connection of the system.

4.5 Lower layer

The development of an accelerator device driver is not planed for this fiscal year. Conventional drivers will be used. The presumption is made that COACK-II is not directly connected with the devices, but is linked with IOC (Input/Output Controllers) and a device controller. In the lower layer, SCADA software, such as LabView and HP-VEE, are also applicable, but are treated the same as those in the upper layer. An intelligent connection, like IOC in the lower layer, is easy to make, but will reduce the network traffic. In other words, adding intelligent or localized control functions to the lower layer will result in an efficient and functionally integrated system.

4.6 Installation of COM and DCOM

The installation of COACK-II is based on the COM building method instead of using a procedural language, or group of library files. Moreover, through the adoption of DCOM, distributed object operation can be kept secure throughout the network.

5 COACK-II COPYRIGHT

In 1999 our application to undertake short, intensive research and development in advanced computational science and technology was accepted by JST. Based on a single fiscal years contract, this project started in April.
KEK signed a research contract with JST, which calls for both organizations to jointly own the copyright. KEK is also cooperating with universities, by drawing up a detailed research and work plan, with the actual job being done at KEK. The component-ware established can be easily downloaded from a web server, with the result that anyone can use it via adaptation to the specific intended accelerator. The right to use this component-ware is now under review in such a way that the joint developers do not need permission to use it.

Joint development and planned users of research labs
Accelerator control is virtually the same in both universities and research organizations, owning to comparable accelerators. This makes it desirable to mutually advance research and development by organizations that have similar systems. Particularly at the evaluation stage of the operational traits of this system, all peculiarities should be clarified through use with many accelerators. In the future, work of this type is expected to advance research activities at our organization.

6 FUTURE PLAN

It is expected that COACK-II will enjoy widespread use as an accelerator control kernel. This will cut the investment required to develop software for accelerator control, make setting up a control system easy, and simplifying the use of the web to globalize operations.
The spread of this system in other technical fields will help promote its general use, which is important in expanding the system. The announcement of this plan drew a great deal of interest not only from universities and research institutes, but also from industrial sectors. Thus, multi-faceted progress in this system is predicted.
We hope that with the creation of this standardized control kernel, so too will come the emergence of many participants and supporters. We also look forward to the global operation of this accelerator, through the introduction of new forms of technology.

7 CONCLUSION

We have analyzed common objects in accelerator domain, and the standard or common objects were defined. These were installed by COM/DCOM as a component-ware on Windows NT. The performances[3] were measured and satisfied. These component-ware are available to down from COACK-II web server. Many SCADA and ACOP are easy to connect using COACK-II, and unified accelerator operation become possible.

ACKNOWLEDGEMENT

The author is indebted to members of VECC, India, and Dr. Nishimura, LBL of USA, for their interest and valuable discussions that contributed to this COACK-II project. The author also wishes to thank Mizuno-san and Kitsunai-san of JST for their substantial support.

REFERENCES

[1] I. Abe, LINAC PC based control system using ActiveX, PCaPAC99 Tsukuba, Japan
[2] http://ninja.kek.jp/COACK2/
[3] T. Nigorikawa, et al., GUI and I/O on COACK-II Project, this conference.
[4] T. Kosuge, et al., Application server and pushing technology on COACK-II, this conference.
[5] P. Duval, Using ACOP in HERA Control Applications PCaPAC99 Tsukuba, Japan
[6] F. Momal, From UNIX to PC based SCADA system PCaPAC99 Tsukuba, Japan