DESIGN OF THE REMOTE DATA ACQUISITION SYSTEM USING JAVA JINI FOR THE 1.8-GEV SYNCHROTRON RADIATION BEAMLINES AT TSRF

Yutaka TAHARA, Noriichi KANAYA, Shoji SUZUKI*) and Shigeru SATO*)
Department of Electrical and Electronics, Faculty of Engineering, University of Ibaraki, Hitachi, Ibaraki, 316-8511, Japan
*) Department of Physics, Graduate School of Science, Tohoku University, Sendai, 980-8578,Japan

Abstract

A remote Data acquisition system has been designed for beamline-control systems for the 1.8GeV TSRF synchrotron radiation source. TSRF is a third-generation synchrotron radiation facility proposed at Tohoku University, Sendai, Japan. The synchrotron radiation beamlines comprise a number of components and sensors to be controlled, including actuators, vacuum valves, radiation-safety interlocks and valve-driving units. The system has been implemented using Java Jini running under the Windows NT. It allows to remotely acquiring control data from these beamline across the computer network.

1 INTRODUCTION

TSRF (Tohoku-university Synchrotron Radiation Source Facility) is a new third generation synchrotron radiation source that is currently proposed at Tohoku University Japan. TSRF is planned to be constructed at the site of Laboratory of Nuclear Science, Tohoku University, where a 300MeV-Linac and 1.2GeV Stretcher Booster Ring are currently in operation for nuclear physics experiments [1]. By taking advantage of the existing facility, TSRF employs the Stretcher Booster Ring as the injector for the TSRF storage ring. This can greatly reduce construction cost for the TSRF generation synchrotron radiation source.

TSRF is designed to provide VUV-SX synchrotron radiation to the experimental hall where experiments such as VUV experiments, surface physics, soft x-ray lithography, microscopy and crystal structure analysis, will be simultaneously carried out. The high-power wiggler/ undulator beam lines are simultaneously in operation, producing very intense synchrotron radiation beams [2]. As shown in Fig.1, the high-power beam lines are distributed along the long circumference of the storage ring. TSRF has a 1.8GeV storage ring with a DBA (double-bend-achromat) type, third-generation storage ring with emittance of 4.9nm·rad, and a circumference of 244m. TSRF has more than ten wigglers/undulators, and thirty beamlines for Soft X-ray and VUV experiments for research.

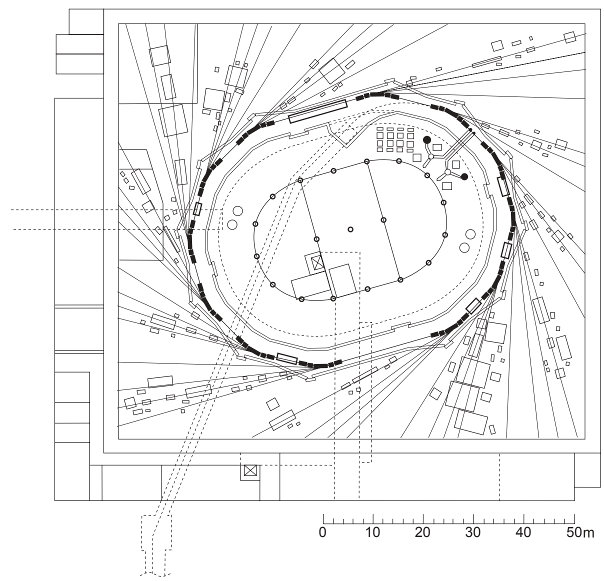

Figure 1: 1.8GeV Synchrotron Radiation Source at TSRF

There are fifty synchrotron radiation beam lines at the TSRF storage ring after fifteen beamlines are constructed the first commissioning phase as shown in Table 1 [2]. These beam lines feed synchrotron radiation to the experimental hall where experiments, such as surface physics, x-ray lithography, microscopy and crystal structure analysis, are simultaneously carried out. These beam lines are simultaneously in operation, providing intense synchrotron radiation beams. The pressures in the storage ring and the beam lines are maintained at an ultra-high vacuum (UHV) of less than 10^{-8} Pa. The beamline are controlled by the distributed control system [3].

A beamline must be controlled independently of adjacent beamlines by the control system in charge of that beamline. In addition, all the beamlines are distributed over the 244-meter round storage ring. Furthermore, there are many experimental users who are not necessary familiar with computer measurement systems for their synchrotron radiation experiments. Measuring instruments and equipment usually have a GPIB (IEEE-488), PCI, VME, VXI and network interface. These are provided by different venders and manufactures. In addition, with these measuring instruments and equipment synchrotron radiation experiments are carried out using different software and operating systems. These aspects impose inconvenient situations where a limited number of facility's beamline engineers must commit these heterogeneous measuring-data-acquisition systems. A new remote-data acquisition system has been designed to systematically control these remote and different beamlines, and allows acquiring data obtained from these beamlines.

Table 1 Wiggler/Undulator beamlines to be constructed for the first commissioning phase at TSRF

Research fields	Source
1. XAFS	Wiggler
2. XAFS (high precision)	Wiggler
3. XAFS (soft X-ray)	Bending M.
4. X-ray diffraction	Wiggler
5. X-ray topography	Wiggler
6. X-ray small angle scattering	Wiggler
7. Atoms and molecules (VUV-SX)	Undulator
8. Surface-solid: PES/fluorescence	Undulator
9. Spin polarized UPS	Undulator
10. Far infrared	Bending M.
11. Millimetre-wave	IR wiggler
12. Soft X-ray optics/microscopy	Undulator
13. Lithography	Bending M.
14. Photochemical process	Bending M.
15. Radiation effect on biomaterials	Bending M.

2 SYSTEM CONFIGURATION

Figure 2 shows the block diagram of the remote data acquisition system for the beam lines. Java programs run on the Virtual Machine (VM) providing homogeneous environment on different platforms independent of their operating systems and hardware architecture. Although the VM runs in principle under any operating systems the VM has no direct interface to physical devices which are tightly implemented upon specific operating systems. Thus those physical devices are not accessible to control applications. The interfaces to physical accelerator components such as digital I/Os, A/Ds and D/As connected to magnets and beam position monitors, are platform-dependent upon each specific operating system. A thin definition interface for PCI modules was written in C for the Java Native Interface (JNI) through which allows Java classes to access the physical devices.

After bootstrapping, a server requests the Lookup server to register its proxy which involves references such as where the server is, a service ID, and all attributes of the server in order to make the methods of the server available to all control programs. The Lookup server is a part of JINI provided by the Java Development Kit or Java SDK [4]. Using the JINI 'discovery-and-join' protocol, a client can find the location of the Lookup server by either sending a multicast message on the network or using a predefined address of the Lookup server, and then asks the Lookup server to locate a server associated with a beamline of which data the client wants to acquire. Upon request from the client, the Lookup server returns the reference of the server's proxy. For the client this proxy is a stub delivered through the Lookup server. The stub is, i.e., the proxy allowing control programs (clients) to invoke remote methods and carries out marshaling of their arguments. There is a transport layer between OS and the physical layer (not shown in Fig.2). It establishes connections and deals with data from/to remote servers. There are more than one Lookup servers for fault tolerant to avoid hang-up when one of them malfunctions during synchrotron radiation operation.

Once communication is established, beamline data can be exchanged among clients and servers by passing arguments of their methods with RMI (Remote Method Invocation) [5]. Unlike the RPC model, it is not necessary to take it into account the binary representation of a data structure of arguments which is a platform dependent factor to be concerned. To transmit an argument data from a server to a client, it is byte-serialized to obtain a byte-stream data, and then transmitted to the client. This mechanism provides a standard way for exchanging beamline data among clients and servers.

In contrast to the pure RMI, a server does not need to bind its name on the registry server which is running as a background process. A client locates where the server is and its related methods are, and it carries out the remote method as if it were a local method to access and obtain data from remote devices to be concerned. Thus any measuring instruments and equipment on the network is transparent to the clients.

Using JINI, a client can access to a server for a certain length of time, so called 'the lease'. The client has to periodically renew the lease by re-accessing the Lookup server. This ensures that the client always keeps the connection to the server even when a network failure occurred or when a system failure occurred. This is also useful for servers when a client accidentally disconnects the connection or shutdown. In such case, the server can disconnect by itself whenever the lease time expires for any reason, and saves its resource for other clients.

A callback mechanism has been implemented for a PCI parallel digital I/O on Window NT. A client does not need to suspend until a remote method completed at the server side. The client can carries out its own process while the remote method is executed. Upon completion of the remote method, the client is notified of it from the server and receives the result. This functionality will be ported to Linux. Implementation of servers for other physical devices such as VME, Serial and GPIB etc is in progress.

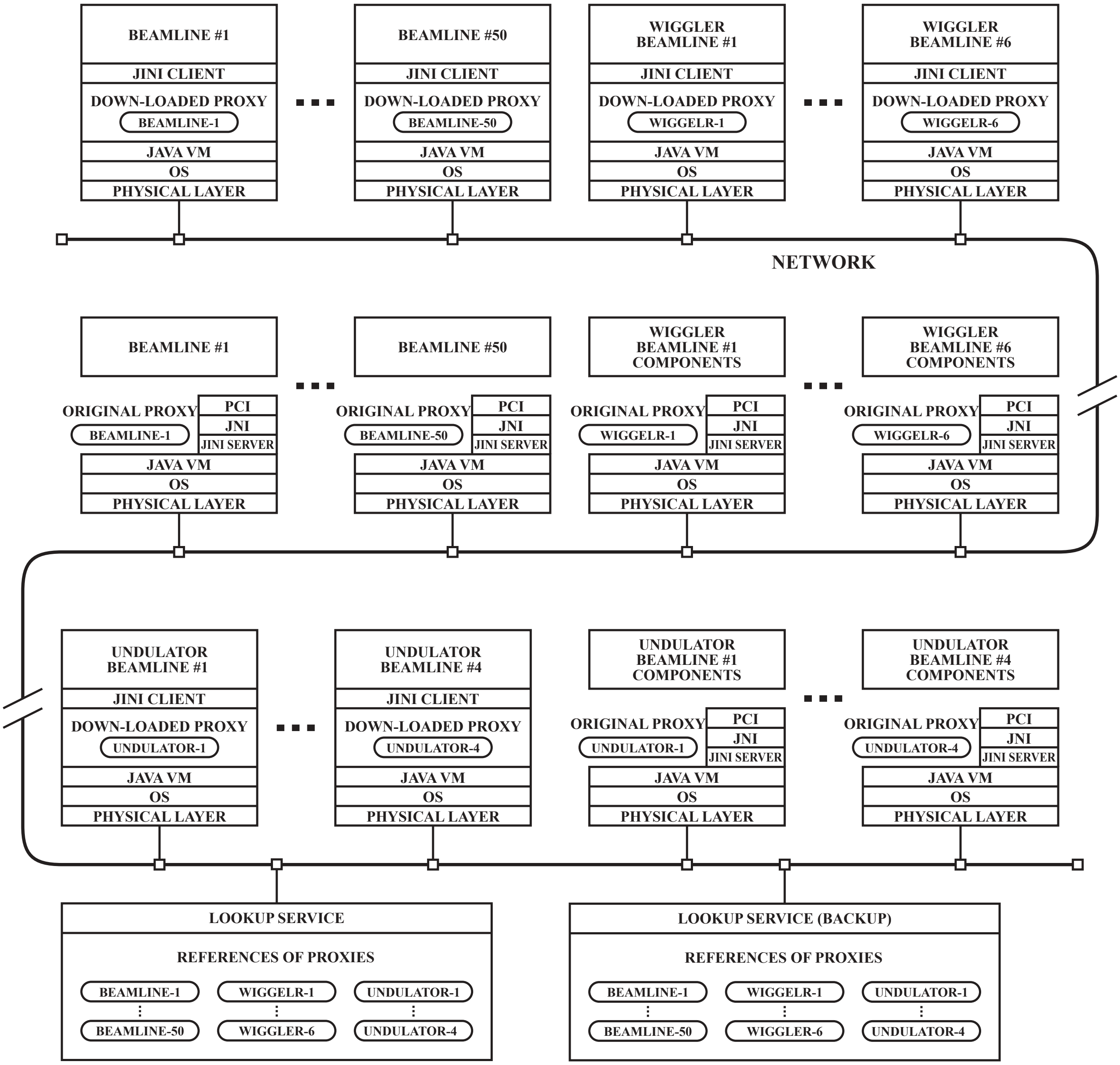

Figure 2: Remote data acquisition system for the synchrotron radiation beamlines at TSRF

3 CONCLUSION

The remote data acquisition system using Java JINI is discussed for the synchrotron radiation beamlines at the TSRF 1.8GeV storage ring. The system deals with PCI modules on remote control systems on Windows NT hosts.

4 ACKNOWLEDGEMENT

The authors wish to express their gratitude to Mr.T.Watahiki, University of Ibaraki for his valuable suggestions, and to N.Kobayashi, Y.Uesugi and K.Murase for their assistance.

5 REFERENCES

[1] M.Oyamada et al.,Proc. of 10th Symp. on Accel. Sci.Tech.p463,1995 (in Japanese).

[2] S.Suzuki, M.Katoh, S.Sato and M.Watanabe, "Design of a Storage Ring Light Source at Tohoku University," Nucl. Instrum. Meth. Vol.A467-468,pp.72-75,2001.

[3] N.Kanaya, Y.Tahara, S.Suzuki and S.Sato, "Remote Equipment as Distributed Remote Objects using Java Remote Method Invocation for the 1.8GeV TSRF Synchrotron Radiation Facility," in these proceedings.

[4] N.Kanaya, S.Suzuki and S.Sato, "Present Status of the Distributed Computer Control System for the 1.8GeV Synchrotron Radiation Source TSRF at Tohoku University," in these proceedings.

[5] *Java JINI*, Sun Microsystems, www.javasoft.com.