LL

3

[REDACTED]

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
CERN - SL DIVISION

CERN LIBRARIES, GENEVA

CM-P00061195

CERN SL/93-42 (CO) ✓

84 3401

9

**THE EQUIPMENT ACCESS SOFTWARE FOR A DISTRIBUTED
UNIX-BASED ACCELERATOR CONTROL SYSTEM**

**Nikolai Trofimov, Serguei Zelepoukine, Eugeny Zharkov, IHEP, Protvino, Russia
Pierre Charrue, Claire Gareyte, Hervé Poirier, CERN, Geneva, Switzerland**

ABSTRACT

This paper presents a generic equipment access software package for a distributed control system using computers with UNIX or UNIX-like operating systems. The package consists of three main components: application Equipment Access Library, Message Handler and Equipment Data Base. An application task, which may run in any computer in the network, sends requests to access equipment through Equipment Library calls. The basic request is in the form Equipment-Action-Data and is routed via a remote procedure call to a computer to which the given equipment is connected. In this computer the request is received by the Message Handler. According to the type of the equipment connection, the Message Handler either passes the request to the specific process software in the same computer or forwards it to a lower level network of equipment controllers using MIL1553B, GPIB, RS232 or BITBUS communication. The answer is then returned to the calling application. Descriptive information required for request routing and processing is stored in the real-time Equipment Data Base. The package has been written to be portable and is currently available on DEC Ultrix, LynxOS, HPUNIX, XENIX, OS-9 and Apollo domain.

*International Conference on Accelerator and Large Experimental Physics Controls Systems
Berlin, Germany, October 18-22, 1993*

THE EQUIPMENT ACCESS SOFTWARE FOR A DISTRIBUTED UNIX-BASED ACCELERATOR CONTROL SYSTEM

Nikolai Trofimov, Serguei Zelepoukine, Eugeny Zharkov, IHEP, Protvino, Russia
Pierre Charrue, Claire Gareyte, Hervé Poirier, CERN, Geneva, Switzerland

Abstract: This paper presents a generic equipment access software package for a distributed control system using computers with UNIX or UNIX-like operating systems. The package consists of three main components: application Equipment Access Library, Message Handler and Equipment Data Base. An application task, which may run in any computer in the network, sends requests to access equipment through Equipment Library calls. The basic request is in the form Equipment-Action-Data and is routed via a remote procedure call to a computer to which the given equipment is connected. In this computer the request is received by the Message Handler. According to the type of the equipment connection, the Message Handler either passes the request to the specific process software in the same computer or forwards it to a lower level network of equipment controllers using MIL1553B, GPIB, RS232 or BITBUS communication. The answer is then returned to the calling application. Descriptive information required for request routing and processing is stored in the real-time Equipment Data Base. The package has been written to be portable and is currently available on DEC Ultrix, LynxOS, HP-UX, XENIX, OS-9 and Apollo domain.

1. Introduction.

The control of an accelerator is based on data exchange with remote devices located all around the accelerator. Today, the heterogeneous mix of accelerator devices has led us to develop a generic equipment access facility. This paper presents a complete software package implementing this generic access facility. This package is implemented and operational¹ in the SPS and LEP controls infrastructure [1] and also in the UNK control project in Protvino[2].

2. Package description.

This software package consists of two libraries (libNEquip.a and libSVMMsg.a) and two background processes (eqname_sv and mh). The libraries allow users to write application software to access their equipment. An Equipment Data Base is created to contain the static information describing the equipment. The following chapters will describe each part of this package.

¹ There are two important uses of the Equipment Server in the SPS-LEP accelerator controls infrastructure. The project of rejuvenating the SPS control system uses the Equipment Server access to implement the migrated Data Modules in the LynxOS PCs [5]. The second usage of this type of access is made in the AT division to build a "bridge" between industrial "automates" and the accelerator world and to provide standard access to industrial PLCs [4].

3. Graphic Overview

Figure 1 Graphic Overview of this package

4. Equipment Data Base.

The Equipment Data Base consists of a GDBM² binary file. It contains all information that is needed by libNEquip.a to communicate with the equipment given its name. An equipment name can have two forms: The first one has been designed at CERN in the 1970's and is made with two 6-bytes strings called family and member. The second form has been newly introduced in this package and consists of a 64 bytes string. For each piece of equipment, the data base contains its localization, its type and its physical address. The localization is the name of the front end computer controlling this equipment. The type specifies which field bus it is connected to or which Equipment Server it is controlled by. And the physical address gives the actual address on the specified field bus. The following example gives some typical lines of this Equipment Data Base:

```
GPIB_1      ... gpsba1 gpib 1 2
MPX_1       ... gpsba2 mpx  2 4 1 2
M1553_1     ... svlsr1 1553 3 4 lock writedata
BBusS15     ... ztls15 bbus 1 48
SYSV_GPIB   ... gpsba3 eqsv ABCD gpib 4 9
SYSV_MPX    ... gpsba4 eqsv MPXQ mpx 1 2 3 4
SYSV_BBUS   ... ztlj56 eqsv BBSQ bbus 1 25
```

Today, this equipment information is extracted from an ASCII file and translated into GDBM binary format. A project of extracting this information from ORACLE RDBMS is under

² GDBM stands for Gnu Data Base Manager which is a simple UNIX data base manager

evaluation at CERN and already used in the prototype of UNK beam transfer line control system. In the near future, this ORACLE extraction will be part of the package.

5. Application Program

The application program is any application software which makes a data exchange with a piece of equipment. It can be written in C or in NODAL³. It makes use of the routines available in the libNEquip.a library of this package.

libNEquip.a basically provides two services. The first is the Equipment name resolution i.e. finding the localization and the characteristics of a named equipment. The second is to exchange data with the equipment. This is done in three steps: building and sending Equipment Message in a network oriented packet to the message handler in the front end computer dedicated to the equipment, waiting for the answer from the message handler and delivering the result to the application program.

6. The Message Handler

This process receives Equipment Messages sent from application programs via libNEquip.a and routes them according to their types. It allows direct access to supported equipment as well as Equipment Server access. It forwards the Equipment Message, waits for an answer and returns this answer to the application program.

7. Equipment

An equipment is a piece of hardware that is accessed by the application program. It can be connected to a field bus or directly to the front end computer. Several types of equipment are supported today by this package. They are 1553, MPX, CAMAC, GPIB and BITBUS. Supported equipment can be directly accessed from the Message Handler. Other types of equipment must be accessed by an Equipment Server.

8. Equipment Server

The Equipment Server is a piece of software written by the user. It runs in the front end computer and directly controls equipment. There are two good reasons to write an Equipment Server: The first one is when the equipment accessed is not a supported equipment. The second is when there are complex actions to be performed on the equipment. A complex action can be made up of several accesses to one or more pieces of equipment. The Equipment Server makes use of the routines available in the libSVMsg.a library of this package to receive and reply to messages coming from the Message Handler.

libSVMsg.a has been designed to ease the writing of Equipment Servers. It hides the communication mechanism with the Message Handler.

³ NODAL is an interpreter that has been designed at CERN in the 1970's which is heavily used for the control.[3]

9. Principles.

This package has been designed to provide simple and generic equipment access. It hides the network and data base communication mechanism. It also supports the old SPS and LEP EQUIP call syntax [6] which has been promoted now to be used for any type of equipment.

Application software in a workstation will access equipment by specifying its name, an action and data to read/write. As the only equipment reference is its logical name, the package will search the Equipment Data Base to find out in which Front End Computer this equipment is physically connected to. An RPC call [7] [8] is then issued to the Message Handler running in this computer which will make the physical equipment call. It is also possible to send the equipment call to a user process called Equipment Server running below the Message Handler. Status or data is returned to the application software.

10. Time Measurements.

These timings have been measured with an application running in an HP/UX 735 running HP/UX accessing a 1553 equipment controlled by a 50Mhz 486 PC running Lynx-OS. A dumb Equipment Server was running in the LynxOS PC.

Data exchange of 1 byte: 30ms for 1553 access, 20ms for Equipment Server access.

Data exchange of 1000 bytes: 50ms for 1553 access, 30ms for Equipment Server access

11. References

- [1] Accessing Equipment in the SPS-LEP Controls Infrastructure: The "SL-EQUIP Package" - P.Charrue - SL/Note 93-86 (CO) - September 1993.
- [2] Control protocols for the UNK control system. A.Elin et al. - ICALEPCS Berlin - October 1993
- [3] The Nodal System for the SPS, CERN 78-07 (CERN yellow report)
- [4] Une passerelle entre le monde des automates programmables industriels et le contrôle des accélérateurs - D. Brahy - AT group Report 93-05 (IC) - Juin 1993
- [5] Understanding Equipment Access in the SPS during the next 6 months - P.Charrue - SL/Note 93-69 - July 1993.
- [6] The LEP/SPS access to equipment manual - The LEP/SPS Controls Group - SPS/ACC/Note 85-23 - LEP controls note 54 rev.
- [7] User Guide to the Network Compiler Remote Procedure Call (NC/RPC) - P.Anderssen, V.Frammery-Paris, G. Morpurgo - LEP controls note 97 - May 1989
- [8] RPC man pages - K.Kostro - SL/CO