THE TECHNICAL DATA SERVER FOR THE CONTROL OF 100 000 POINTS OF THE TECHNICAL INFRASTRUCTURE AT CERN

P. Ninin, H. Laeger, S. Lechner, R. Martini, D. Sarjantson, P. Sollander and A. Swift
CERN, Geneva, Switzerland

Abstract

We have defined a Technical Data Server (TDS) to be used for the supervision and control of the technical infrastructure of CERN by its dedicated control room, the Technical Control Room (TCR), and by equipment groups. The TDS is basically a real-time information system which contains all states of the technical infrastructure: 100 000 points describing electrical distribution, cooling water, air-conditioning, vacuum, safety, and similar systems. It is expected that the TDS will substantially increase system performance and ease operation. As the concept of such a data server is also of interest to other groups in CERN accelerator and experimental physics divisions, detailed software and user requirements, as well as criteria for selection, implementation and maintenance have been elaborated in common with these groups. The project adheres fully to the European Space Agency (ESA) PSS-05 Software Engineering Standards and its life-cycle approach. This paper describes the data acquisition and distribution mechanisms, the interfaces to equipment and to existing alarm and data logging systems and to operator supervision consoles, the alarm reduction mechanism, and the error-handling and logging. Problems encountered during the project development are discussed in some detail.

I. INTRODUCTION

At CERN, the Technical Control Room (TCR) monitors data coming from the electrical distribution, cooling water, air conditioning, vacuum, cryogenics, safety, and other systems. In this context, a Technical Data Server (TDS) has been defined and will provide data collected from the above equipment to high-level control software such as Human–Computer Interfaces (HCIs), the logging system and the alarm server. An expert system will be used to perform alarm filtering.

This project is being developed using the European Space Agency (ESA) PSS-05 Software Engineering Standards [1].

The intention of this paper is to highlight the scope of the TDS project, the project environment and how we intend to achieve the project goals (methods, tools). There will also be a detailed description of the chosen middleware package, RTworks by Talarian.

II. MOTIVATION FOR THE PROJECT

The project involves a redesign of the existing methods of acquiring and managing data used in the TCR. At present, data for the HCI and logging is polled from the equipment directly (top-down) and alarm data is handled by event-driven software (bottom-up). Data is acquired through a large distributed network that covers all accelerator sites: PS, SPS, LEP, Meyrin and Prévessin.

The benefits of the TDS are numerous. It keeps a permanent image of all equipment attributes monitored by the TCR (100 000 points). It proposes a new, more reliable channel for alarm transmission to cope with alarm bursts and a centralized alarm reduction mechanism. The response time of the Uniform Man–Machine Interfaces (UMMIs: applications which display equipment states), is improved, to respond to data requests within one second. Equipment access is rationalized, as specific access routines are replaced by a generic addressing mechanism which allows both data retrieval and sending of commands. The use of an industrial middleware package decreases the maintenance effort for multiple client-server applications. The number of processes in each hardware element is decreased, due to the fact that multiple individual equipment access no longer takes place. Finally, the TDS offers a solution to the problem of supplying data to the increasing number of TCR applications users. The TDS provides high availability and reliability (24 hours a day, 365 days a year) as required for the operation of CERN services. All data monitored by the system are defined in a unique reference database and identified by a unique tag [2].

III. THE ENVIRONMENT

The environment consists of a three-layer architecture [3]: the Control Room layer, the Front-End Computing layer, and the Equipment Control layer (Fig. 1). The Control Room layer consists of HPUX servers (HCI, alarms, RDBMS) and X-Terminals. The HCIs are based on an XWindow–OSF/Motif user-interface management system and Dataviews [4]. The Front-End Computing layer consists of Front-End process computers (FEs), also called Process Control Assembly (PCA), based on PCs and VMEbus crates. They interface with various fieldbuses and more particularly the MIL-1553 fieldbus.

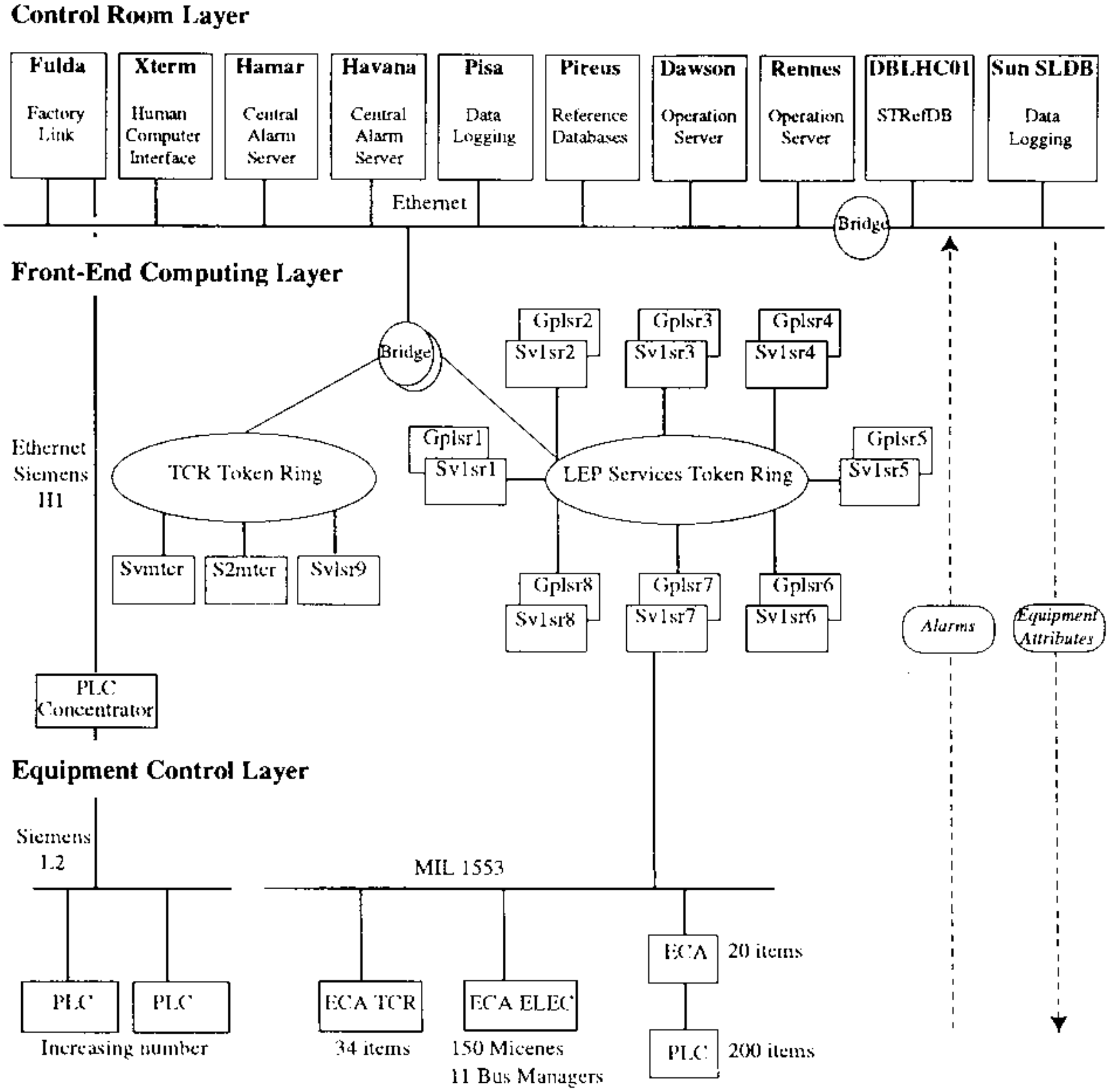

Figure. 1. The TCR Control Environment

The communication between the applications used in the Control Room layer and the Front-End layer is achieved through Remote Procedure Calls. The new generation of FEs consists of a standard VME rack powered by a PowerPC processor and running LynxOS. The Equipment Control layer consists of Equipment Control Assemblies (ECAs) connected to the FEs via various equipment fieldbuses (MIL-1553, GPIB, BITBUS, JBUS) or via RS232/422 links. At some places the ECAs provide the interface to local control systems, such as Landys & Gyr or to programmable logic controllers (PLC).

Network communication is provided by local Ethernet segments bridged to large token-rings. There is one specific token-ring network for the LEP services. These networks will be replaced by a 100-Mbit/s FDDI backbone that will cover the entire CERN site.

The hardware environment for the target system will consist mainly of two types of machine. The first is the HP700 series which will execute the archive logging, logical analysis and data distribution for the information system. The second platform is the VME-bus PowerPC's 603/604 which will execute the data access module.

It is expected that the TDS will run under the HP-UX UNIX operating system, XWindows, X11 and LynxOS. All XWindows application development will adhere to the OSF/Motif standard.

IV. ACHIEVING THE GOAL

In order to achieve the proposed goal, the TDS project team adopted the ESA PSS-05 Software Engineering Standard [1]. After following the Standard through its first phase, the User Requirements phase, and into the second phase, the Software Requirements phase, a detailed logical model was created and a market survey was performed.

The market survey, which considered many middleware products, was conducted in order to determine whether the project could be implemented using a commercially available product. The investigation resulted in a comprehensive document that detailed each of the middleware packages most likely to satisfy the logical model. One particular package, RTworks by Talarian, was highlighted as being the product that would meet our needs.

V. MIDDLEWARE

A. Technical overview

RTworks is a suite of software development tools for building time-critical monitoring and control system applications. It consists of separate processes for data acquisition, data distribution, real-time logical analysis and graphical user interfaces. RTworks is specially designed for building applications where large quantities of data must be acquired, analysed, distributed, and displayed in real time. The RTworks processes communicate via a dedicated message server. Applications can run on a single workstation or can be distributed across multiple processors in a heterogeneous network.

The RTworks client-server architecture is built specifically to offer high-speed inter-process communication, scalabilty, reliability, and fault tolerance. As the needs of the application grow RTworks, user-development, or third-party software processes can be added transparently.

The major RTworks software processes are:

RTserver - information distribution server
RTie - expert system builder and inference engine
RThci - dynamic graphical user interface builder
RTdaq - data acquisition interface
RTarchive - intelligent information archiver
RTplayback - information playback module.

Tags describing similar equipment are gathered in 'datagroups'. Client applications subscribe to these datagroups and receive all related tag changes.

The design of RTworks fits very nicely with the logical model defined for the TDS. This likeness was further highlighted by the fact that the RTserver provides a real-time image to all of its clients showing the current state of the hardware from which it is receiving data. This immediately fulfilled one of the most important TDS requirements.

With the RTie module installed, complex analysis can be performed on data within the TDS, which, when using a distributed system, is at relatively little processing cost. This again proved to fulfil a key project requirement.

Data can be stored for later analysis using RTarchive and can be retrieved using RTplayback. RTplayback uses a standard relational command language (similar to that of SQL) and allows remote command processing, again via any of the common network communications protocols. The technical specification is summarized in Table 1.

TABLE 1: RTworks - Technical Specification

Minimum hardware	*CPU*	All current WS
	Memory	All current WS
	Platform	All current WS
Version	*Development*	YES
	Runtime	YES
Software	*O.S.*	UNIX,VMS
	Programmer's toolkit	YES
	Language	C
Network	*Interface I/O*	User-developed
	IBM TR	NO
	Ethernet	YES
	TCP/IP	YES
	Interface to PC and Macintosh	YES
Support	*Hot line*	YES
Reliability	*System redundancy*	YES
	Diagnostic tool	YES
RTdb	*Maximum no. of tags*	Unlimited
	Update rate	User-written daq
	RAM resident	NO
	SQL compatibility	YES
	No. of task access.	Publish /Subscribe
	Tag reserved by system	NO
	On-line update	YES
	Tag timestamp	YES
	Tag validity flag	YES
	Maximum tag no. retrieval	Unlimited
	Tag query average	User-dependent
	Access protection	YES
Alarming	*Maximum no. of tags*	Unlimited
	Masking	YES
	Alarm severity class	User-defined
	External interface	YES
	Alarm reduction	Inference engine
	Access to reduced alarm	YES
HCI	*Maximum no. of users*	Licence based
	Graphical animation	Dataviews
	X11R5/MOTIF	YES
	Interface to external HCI	YES
	Run time licence	YES
Event detector	*Maximum no.*	User-dependent
	Minimum time interval	User-dependent
	Polling	YES
	Event-driven	YES
Maths & logic	*Interpreted*	Inference engine
	Compiled	NO
	Loop	YES
	Conditional statements	YES
	Block structure	YES
	Data filtering	YES
Trending	*Real-time*	YES
	Maximum no. of curves	Graph-type dependent
	Tag change logging	YES
Data logging	*Interface with ORACLE*	YES
Administration	*Configuration*	YES
	Error reporting	YES

Having thoroughly analysed all the alternatives, we are convinced that RTworks provides the quickest and most cost-effective long-term solution to this project.

Figure 2 shows how RTworks fits into the TCR control environment.

VI. THE TDS AND ITS INTERFACES

A tag is identified by a generic format that describes one point of data monitored by the TDS. The tag semantics is based on an object-oriented approach; it refers to the class of equipment, a specific member within that class, and a specific attribute of the class. When the RTdaq modules receive Dynamic Point Information (DPI) from the equipment interface they convert this information into tags. These tags can belong to one or more datagroups. The RTdaq modules pass these tags to the RTserver which distributes them to the client applications (RThci, RTie, user clients). The inference engine analyses the data it receives and performs actions upon it. The server distributes the tags to all interfaces that subscribe to the datagroup that the tags belong to. Only the real-time image of the equipment is available from the server. Newly added interfaces, such as an application that has just been started, receive the current tags for all the datagroups that they subscribe to. All tag changes sent to the TDS are logged in a 72-hour ring buffer. Tags are time-stamped as closely as possible to the time at which the hardware generates the values. The TDS distributes a standard time in order to synchronize all equipment generating values. Archiving is achieved using the RTarchive module and the RTplayback module is used for data retrieval from the archive.

The TDS interfaces with the following TCR and equipment group systems [5]:

- the equipment control systems
- the Reference Databases (STRefDBs)
- the Central Alarm Server (CAS)
- the HCI applications
- the data and event logging systems
- the TDS Administrator
- the PC/Macintosh environment.

The operations expected of these interfaces will be described in the following sections.

A. *Equipment control systems*

The Technical Data Server initially interfaces with four types of equipment control systems:

– The MICENE TMS99-95s which have been used since 1984, 150 such units are installed. They control the CERN electrical distribution.

– The BUS Managers are based on a VME rack and powered by a 68030 microprocessor running under OS9. They interface with several industrial GBUSes and 11 such units are currently installed. The BUS Managers are the new generation of MICENE.

– ECATCRs, 34 installed, handle miscellaneous data for the TCR. Each uses a VME bus powered by a 68000 under OS9.

– Industrial systems such as Siemens Programmable Logic Controller (PLC) using the Sinec H1 protocol and Landys & Gyr VISONIK.

In the future there should also be interfaces to the vacuum and cooling systems where a specific protocol is currently used.

The TDS supports both event-driven and polling data-acquisition mechanisms. The acquisition of data from the hardware level is made using specific device drivers which convert the heterogeneous data coming from the various sets of equipment into tags which are handled by the RTdaq. These modules are located on Process Control Assemblies (PCAs). Equipment drivers and RTdaq modules exchange data according to a unique equipment access protocol. This protocol is implemented using TCP/IP sockets. The tag value is stored in a shared database within the RTdaq.

Industrial equipment is capable of sending alarms that can generate automatic commands such as calls to paging systems. These alarms are known as Automatic Triggered Output (ATO). The system that manages the ATOs will be integrated into the TDS architecture.

B. *Reference databases*

An ORACLE reference database (STRefDB) contains all static and logical descriptions of the points monitored by the TDS in the form of tags. The tags are described in terms of: physical address, description, type, class, member, generated alarm level, etc. Upon initialization and maintenance of the TDS, tag definitions are down-loaded from the STRefDB database to configure the information system.

C. *Central Alarm Server*

The Central Alarm Server (CAS) is a key software element in TCR operation. It centralizes the alarms generated by the technical infrastructure. Recent investigation has shown a bottleneck in the alarm transmission channel; hence the current transmission system is being reviewed as a part of the TDS project. For the electrical system alarm reduction is currently performed within each PCA, before the alarm is sent to the CAS. This reduction will be moved to the TDS level, since a higher degree of reduction can be performed there. Indeed, TDS will know the current state of all electrical equipment, not only the state of the equipment connected to a single PCA. The TDS will be integrated into the Central Alarm Processing Environment (CAPE).

D. *UMMI applications*

The UMMI applications consist of the software used by the TCR to survey the equipment. They give a graphical representation of the controlled processes and are animated by data directly acquired from the equipment (polling mechanism). They allow users to send commands to the hardware and they are also used by equipment groups. Some of the applications consist of more than 100 views. The response time is particularly affected by the number of points acquired, the number of PCAs and ECAs involved, and their load. The TDS should improve this situation by offering a rapidly accessible image of the current state of all equipment. User applications will access the TDS with generic C routines that subscribe to the required datagroups. All user commands will be logged. These applications will become event-driven.

E. *Data and event logging systems*

The Data Logging System (DLS) [6] accesses various types of equipment: electrical, ventilation, water systems, safety, etc. It interfaces with them using a number of logging processes making calls to the SL-EQUIP package [7]. The measurements are performed at a frequency defined in STRefDB with the fastest rate being every 60 seconds. The advent of the TDS will have a large impact on the current logging system.

The connection between the data logging system and the TDS will be very similar to the connection with the UMMIs. Generic C routines will be used to access the tags required. The data logging may also act as a subscriber to tags to be logged.

The Event Logging System (ELS) is still under analysis. Its main purpose is to record events, not declared as alarms, generated by the technical equipment. The system will be able to capture the events at any level of the control system and store them for further analysis. This system may be integrated directly into the TDS architecture.

F. *TDS Administrator*

This application should offer tools to monitor all internal TDS activity as well as the state of its interfaces. The TDS Administrator will handle, amongst other things, application admissions to the system, tag unavailability, system monitoring, and error identification and recovery.

The TDS Administrator should help to ensure that the TDS will be available 24 hours a day, 365 days a year. Good error reporting and protected access to the TDS will provide improved safety in the working environment. No loss of data should occur in the TDS. This implies the installation of a hot-spare system. The monitoring of the TDS will include the monitoring of its data-acquisition modules which could be widely distributed.

G. *PC and Macintosh environment*

TDS data will be made available on both PC and Macintosh platforms and could use the promising features of Web browsers. The effective use of the Web will be investigated during the design stage of the TDS. An alternative would be to use a DDE socket. Currently TCR UMMI applications are distributed on PC and Macintosh using Xvision or Mac X.

VII. CONCLUSION

The integration of RTworks into the CERN technical infrastructure will be gradual. The TDS project team should have prototyped the basic functionalities of the overall system by the time this paper is released. The prototype, using RTworks, should include scaled versions of:

- the interface to the STRefDB
- the interface to the equipment
- the alarm reduction
- the CAS interface
- the data logging
- the HCI
- the TDS Administrator.

Approximately 2000 points from three electrical substations will be managed by the prototype. This should indicate which areas of the system will require more work than others and highlight deficiencies in the specifications. Unitechnic-France have committed themselves to the porting of the RTworks data acquisition module to the PowerPC platform running LynxOS.

In accordance with the CERN outsourcing policy, the prototype is being developed in collaboration with STERIA, a company experienced in the field of RTworks integration.

The TDS will change the distribution of responsibility in the monitoring of the CERN technical infrastructure. The TDS will impose a strategy and a common structure for equipment data-acquisition, treatment (alarm, logging), transport and static definition. It will guarantee the distribution of the information to all its client modules. The TDS will provide a common interface to the equipment in order to replace all equipment-specific data servers. CERN equipment drivers and industrial drivers such as the SINEC H1 will exchange data through a standard UNIX communication mechanism with generic RTdaq modules. Nevertheless, the driver that accesses the equipment will stay under the responsibility of the equipment groups.

The TDS adopts a new control system philosophy based on the asynchronous distribution of technical equipment attributes. All TDS components are optimized to provide high performance and reliability.

The task ahead is not an easy one, indeed, it requires a redefinition of a structure that has worked for many years. With new projects such as the Large Hadron Collider on the horizon the technical infrastructure of CERN will have to handle larger quantities of data.

References

[1] “Software Engineering Standards”, ESA Board for Software Standardisation and Control, 1994.

[2] D. Sarjantson, S. Lechner and P. Ninin, “Technical Data Server Software Requirement Document”, (ST internal document, April 1995).

[3] M. Vanden Eynden, “The New Generation of PowerPC VMEbus Front-end Computers for the CERN SPS and LEP Accelerator Control Systems”, CERN/SL 95–74, August 1995.

[4] P. Ninin and P. Sollander, “Utilisation d’un logiciel commercial graphique pour le contrôle des installations techniques du CERN”, CERN–ST–MC–TCR/92, October 1992.

[5] D. Sarjantson and P. Ninin,“Technical Data Server, Interface Control Document”, (ST internal document, June 1995).

[6] R. Martini, H. Laeger, P. Ninin, E. Pfirsch and P. Sollander, “Data Logging for Technical Services at CERN, these Proceedings

[7] P. Charrue, “Accessing Equipment in the SPS–LEP Controls Infrastructure: the SL-EQUIP Package”, SL/Note 93–86 (CO), September, 1993.

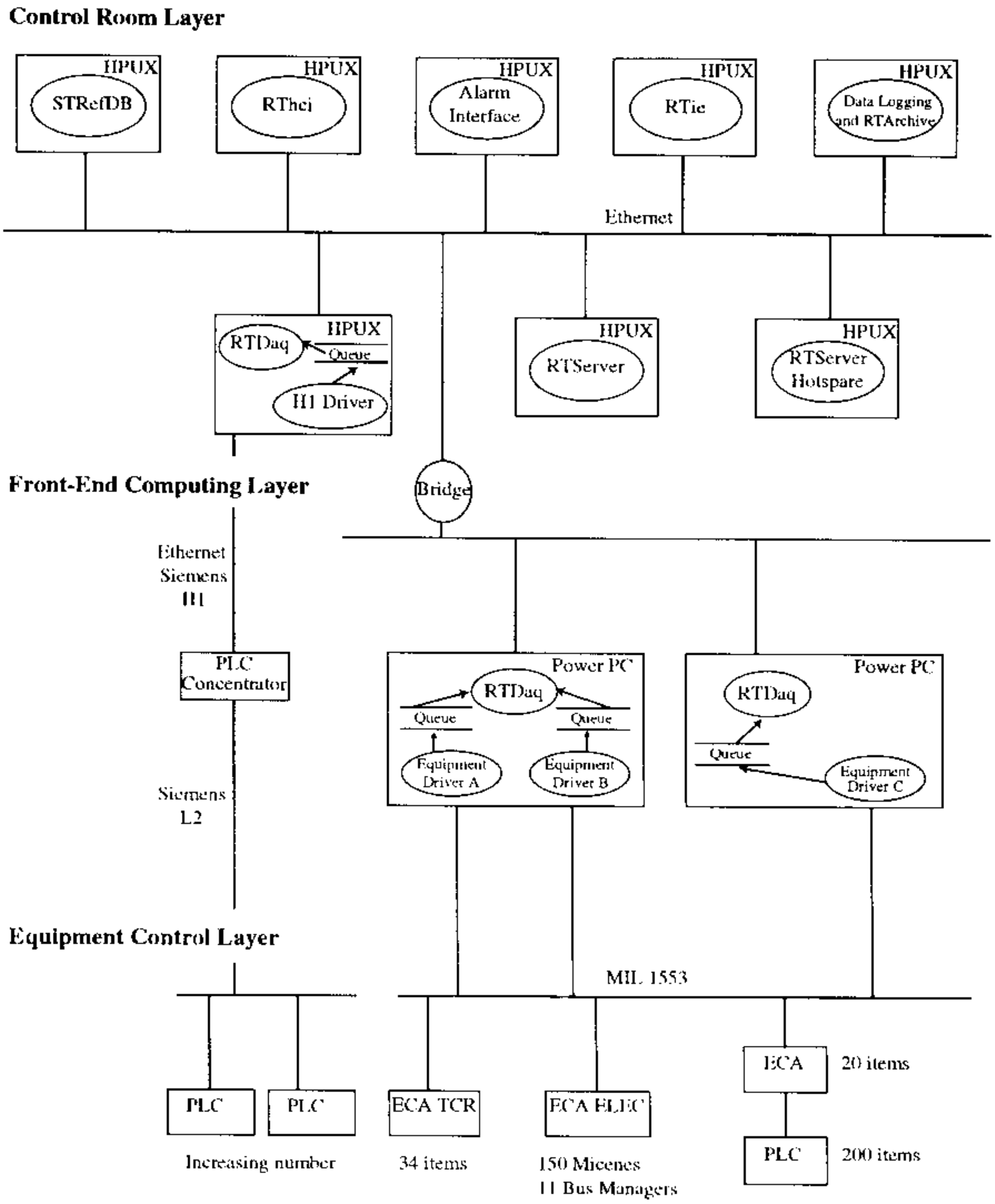

Figure. 2. The Proposed RTworks Structure