Intern. Federation of Automatic Control (IFAC) Workshop on Distributed Computer Control Systems, Tampa, Florida, October 2-4, 1979.

BNL 26731

MASTER

THE ISABELLE CONTROL SYSTEM - DESIGN CONCEPTS*

CONF-7910170--1

Dr. J. W. Humphrey

Brookhaven National Laboratory
Upton, NY 11973

Abstract. ISABELLE is a Department of Energy funded proton accelerator/storage ring being built at Brookhaven National Laboratory (Upton, Long Island, New York). It is large (3.8 km circumference) and complicated (~ 30 000 monitor and control variables). It is based on superconducting technology. Following the example of previous accelerators, ISABELLE will be operated from a single control center. The control system will be distributed and will incorporate a local computer network. An overview of the conceptual design of the ISABELLE control system will be presented.

Keywords. Centralized Plant Control, Computer Control, Process Control, Microprocessors, Minicomputers, Particle Accelerators.

I. GOALS OF THE ISABELLE CONTROL SYSTEM

ISABELLE is a proton accelerator/storage ring. From the point of view of control system designer, it is large (3.8 km circumference) and complex (approximately 30 000 monitor and control variables). ISABELLE's function is to provide proton-proton collisions for the use of elementary particle physics experiments. In other words, ISABELLE is a utility. Thus the goals of ISABELLE's control system are more akin to those of the control system of a public utility or petrochemical plant than they are to the goals of the data acquisition and control systems of the avant garde experiments ISABELLE serves.

ISABELLE comprises a number of systems. There are approximately 1000 superconducting magnets cooled by a large cryogenic refrigeration system and powered by over 300 power supplies. Beams of up to 8 amperes of protons in each of two rings circulate in a beam pipe evacuated by a large distributed vacuum system. There is an injection system which transports protons from the present Alternating Gradient Synchrotron into ISABELLE. A stacking rf system is used in the accumulation of beam current and an accelerating rf system accelerates the protons up to 400 GeV. The beams are stored for periods of one to two days, providing proton-proton collisions during that entire period. There is an extraction system which ejects the remaining protons at the end of the useful storage time, and transports them to a beam dump. The control of these various systems is unified by the ISABELLE control system - a local network of distributed minicomputers communicating with devices via a process data highway.

The goals of the ISABELLE control system can be simply stated:

1. All ISABELLE systems will be operated from a single control center.

2. In the case of the loss of all or part of the control computer network, accelerator devices will continue to operate at the most recent setpoint.

3. The hardware and software interface of devices to the ISABELLE control system will be implemented in standard ways.

4. The generation of programs and the testing of new devices should be routinely done by control system users.

5. The control system should be capable of phenomenologically modelling the state of ISABELLE - both in a predictive and historical sense.

A. Discussion of the Control System Goals

1. All ISABELLE systems will be operated from a single control center. This is by now standard practice in accelerator control systems. The existence of local control rooms, dedicated to a single system, is not incompatible with this goal. However, any computer control capability available in a local control room should also be present in the ISABELLE control center.

2. In the case of the loss of all or part of the control computer network, accelerator systems will continue to operate at the most recent setpoint. In addition, such a loss will not, of itself, cause the beam to be aborted. This principle or goal has important implications for the control system design. A

---

*Work performed under the auspices of the U.S. Department of Energy.

necessary condition for its achievement is that no feedback loop be closed through the control computer network. Such a goal has only becomes feasible because of the dramatic drop in the price to performance ratio of digital electronics. DDC (direct digital control) loops can be implemented in the devices which need them. The control system need no longer be involved in softwear servo loops. In the classic control engineering terminology, ISABELLE's control system is an open loop system.

In actual practice, this goal will be ignored from time to time. The feasibility of digital feedback loops can certainly be tested by using the control computer network. Once the feasibility has been proven however, such loops will be implemented in dedicated hardware.

3. The hardware and software interface of devices to the ISABELLE control system will be implemented in standard ways.

The ISABELLE control system can itself be considered a utility. It supplies a digital communications path and common computer facilities. Devices connected to the control system must satisify appropriate connection standards for both hardware and software. The importance of protocols and interface standards is well known in the distributed data processing field. This is conceptually a similar problem. Consider the connection of a new type of magnet power supply to the control system. The power supply must have the appropriate plug through which the appropriate pulse trains are transmitted to communicate with the appropriate softward driver and data base module in the local computer node. The same communications hardware is used by many devices and device designers, hence the need for hardware standards. Once a device is incorporated in the control system, the device associated software such as device drivers and data base modules can be used by many programs and programmers, hence the need for software standards.

4. The generation of processes and the testing of new devices should be routinely done by the control system users.

It has always been true that one of the persons who is more knowledgeable than most about the capability of a device is the device designer himself. The SPS took advantage of this truism by implementing an easily learned computer language (NODAL) which allowed the device designer to become the applications programmer who used the device. The developments that have taken place since the SPS control system was designed have increased both the need and the opportunity for this amalgamation. The need has increased because the use of microprocessors and other LSI (VLSI) techniques has increased the sophistication and capabilities of the devices themselves. The opportunity has increased because the use of imbedded microprocessors has increased the level of sophistication of the software abilities of the device designers.

In most cases, the testing of new devices cannot be done during production running, when ISABELLE is producing luminosity for the use of elementary particle physics experiments. Thus the testing of devices in the realistic accelerator operations environment will occur at infrequent times and for restricated periods. To maximize the effectiveness and efficiency of the use of these periods, the device designer is the appropriate resource of which to take advantage. He must have at his hand comprehensive debugging and monitoring facilities. It is worthwhile noting the common experience that interpretive computer languages (BASIC, NODAL) are very useful in the testing and wringing out phase of device construction.

5. The control system should be capable of phenomenologically modeling the state of ISABELLE - both in a predictive and a historical sense.

Experience with SPEAR has shown the accelerator community that it is possible to control the global parameters of an accelerator, rather than the individual specific control elements. Such operation is only possible through the use of a mathematical model of the accelerator which exists in a computer of the control system. This model transforms the changes in accelerator parameters into increments of currents in power supplies, phase and amplitude of the rf accelerating system, etc. If the changes are so large that they can only be accomplished by a series of incremental changes, then the model calculates the increments, and then implements the series under open loop software control. Such global control is clearly a useful feature. After all, it is an accelerator that is being controlled, and it stands to reason that the language of the accelerator (beam energy, beta function, crossing angle, chromaticity, etc.) should be used, rather than some more arcane engineering language (volts, amperes, gauss, etc.).

For ISABELLE, this feature is more than useful, it is necessary. ISABELLE is complex and fragile. An operator needs to know the consequences of actions before the actions are implemented. If a beam manipulation will cause the beam to exceed the dump profile, the operator has to be informed of that fact before the manipulation is performed. Such proposed actions can be tested in the accelerator model, and the implications can thus be made known to the operator. As the predictive accelerator model is refined by operations experience, it will become an indispensible element in the operation of ISABELLE.

The historical model of ISABELLE is also a critical necessity. It will happen that the accelerator will be running and all at once, an alarm will be actuated and there will be no beam in the machine - because the beam extraction system will have been triggered. What has happened? Certainly the option to go back and do it again must only be used as a last resort - especially if quenching superconducting magnets is involved. It must be possible to phenomenologically reconstruct the state of ISABELLE and analyze what was the sequence of events which lead to the beam extraction. There are two requirements for this reconstruction process. The first requirement is data. Since ISABELLE is so complex (See Table 1.), and since many variables can change rapidly, the amount of data associated with such an event might be quite large - of the order of 5 megabytes, for example. In addition, since the occurrence of such events is by definition unpredictible, the data acquisition facility must be constantly operating. The second requirement for the reconstruction process is a model/procedure to analyze the data associated with such an event; i.e., the historical ISABELLE model.

II. INTRODUCTION

The basic design concept of the ISABELLE Control System is that it will be a distributed system. This conclusion is based on ISABELLE's size and complexity. The rationale for this conclusion has been well put by the authors of a design report which came to a similar conclusion in the case of the control system for a large steel manufacturing plant:

"Automatic control of the modern steel mill, whether achieved by a computer based system or by conventional means, involves an extensive system for the automatic monitoring of a large number of different variables operating under a very wide range of process dynamics. It requires the development of a large number of quite complex, usually nonlinear, relationships for the translation of the plan variable values into the required control correction commands. Finally, these control corrections must be transmitted to another very large set of widely scattered actuation mechanisms of various types which, because of the nature of the steel manufacturing processes, involve the direction of the expenditure of very large amounts of energy. Also, plant personnel, both operating and management, must be kept aware of the current status of the plant and of each of its processes.

Such a [distributed] system should have many benefits that can be of great value when installing, operating, or altering the system. Some of these benefits are:

1. Flexible system configuration - distributed subsystems may be modified, replaced, or deleted without upsetting the rest of the system.

2. Graceful degradation - failure in one or more components or subsystems will not cause the entire system to fail.

3. High systems reliability due to:

   a. Easy to add parallel redundant units and subsystems which can be incorporated to back up and duplicate the functions of the main components and subsystems.

Table I. There is a total of 21184 monitor variables and 10364 control variables in ISABELLE.

Device	System	Number
Monitor Variables		
Coil Voltage Dipole	Magnet	2928
Coil Voltage Quad	Magnet	2712
Magnet	Magnet	300
Orbit Position	Beam Monitoring	444
Lead Voltage Drop	Cryogenics	1024
Lead Temperature	Cryogenics	612
Pressure Transducers	Cryogenics	72
Flowmeter	Cryogenics	48
Magnet Temperature	Cryogenics	456
Gauges Ins. Vac.	Vacuum	324
Gauges UHV	Vacuum	1416
Ion Current Pump	Vacuum	1416
Bakeout Temperature	Vacuum	2832
Sector Valves	Vacuum	84
Turbopumps	Vacuum	158
Clear Electrodes	Vacuum	1416
Turbopumps Ins. Vac.	Vacuum	132
Gate Valve Ins. Vac.	Vacuum	648
Radiation Monitor	Access	162
TI Pump Current	Vacuum	1416
Instrumentation	Beam Monitoring	24
Temperature Sensors	Cryogenics	150
Pressure Transducer	Cryogenics	150
Flow Heater	Cryogenics	12
Vibration Analyzer	Cryogenics	36
Impurity Analyzer	Cryogenics	2
Digital IO	Cryogenics	500
Transfer Syst. Total	Injection	1000
Ejection Syst. Total	Ejection	500
Cavities	Radio-frequency	40
High Level rf	Radio-frequency	120
Low Level rf	Radio-frequency	40
Control Variables		
Power Supply 300 A	Power Supplies	144
Power Supply 100 A	Power Supplies	96
Power Supply 50 A	Power Supplies	252
Lead Flow	Cryogenics	612
Magnet Flow	Cryogenics	12
Lead Voltage	Cryogenics	1024
Valve Control	Cryogenics	48
Sector Valves	Vacuum	84
Ion Pumps On/Off	Vacuum	1416
Ti-Pumps	Vacuum	1416
Bakeout Heater	Vacuum	2832
Turbopumps UHV	Vacuum	84
UHV Gauges On/Off	Vacuum	1416
Turbopumps Ins. Vac.	Vacuum	66
Gate Valves Ins. Vac.	Vacuum	324
Power Supply 5 kA	Power Supplies	2
Power Supply 300 A	Power Supplies	10
Power Supply 50 A	Power Supplies	16
PID Control Loops	Cryogenics	60
Digital Control	Cryogenics	200
High Level Analogue	Radio-frequency	70
Low Level Analogue	Radio-frequency	80
High Level On/Off	Radio-frequency	60
Low Level On/Off	Radio-frequency	40

b. Transmission of partially processed plant information allowing:
   (i) Decreased data rates since processors are distributed to functional areas and only processed information need be sent between any two subsystems rather than raw data as formerly.
   (ii) Use of error detection codes which allow any fault or casualty condition in the system to be detected and identified by the processor in its area of responsibility.

4. Lower cost due to:

a. Simplified hardware configuration packaging since processors need not be large due to the reduced processing requirements of each processor.

b. Simplified software because functions are carried out by several small, locally responsible processors, not by a large machine that must perform all of the control functions and calculations within the entire control system.

c. Large scale integration technology.

d. Multiple use of standard components. Many different subsystems can use identical hardware to perform varied functions.

e. Ease of incrementally increasing capability since units may be added to the system without drastically interfering with the functions of the rest of the system.

f. Simplified installation since common data channels can be used for processor to processor communication. This eliminates the need for individual multiple-wire cables between any two units." (Purdue University Project Staff 1977).

What are the various elements of the distributed system used to control ISABELLE? To answer that question from a system viewpoint, consider the most common task that the control system will be called upon to accomplish, namely to allow an operator to control or monitor a device. (See Fig. 1.). In accomplishing this task, the operator normally intersects with the system at a control console, from which he can cause a program or command to be executed. Information concerning this command is communicated from the control console computer to a local computer over the computer network communication system. Resident in the local computer is a piece of software called a data module. The command sent to the local computer informs the data module what the operator wants. The data module communicates over the process data highway to the device. The command is executed by the device, and information of interest may be passed back through the system to the control console and communicated to the operator. All these communications and their corresponding protocols and interfaces are transparent to the operator. The operator thinks he is communicating directly with the device.

Thus, the control system can be conceptualized as a control center (control console/computer), a computer network, and a process (or device) interface system. Each of these three elements will be considered in more detail in the following pages.

III. CONTROL CENTER

The Control Center is where operators interact with ISABELLE qua system. The Control Center is composed of a number of elements:

1. Control Consoles.
2. Data base Facility and Program Library.
3. Analytic Computation Facility.
4. Program Development Facility.
5. Alarms and Access.

1. Control Consoles: In the language of the control engineer, a control console is known as the man/machine interface. The International Purdue Workshop on Industrial Computer Systems postulates two subsystems that make up a control system. They are the personnel subsystem (operator, management) and the machine subsystem (computers, controllers, industrial plant or process, etc.). "The Man/Machine Interface (MMIF) is that boundary between the two subsystems across which information and control manipulation flows. In physical terms, it is typically the face of a console which contains displays and keyboards of various types. The hardware and software behind this boundary participates in translating human inputs to control signals and process inputs into data displays for humans." (MMIF Committe 1978).

At ISABELLE, the control console (MMIF) will be a computer. Accelerator experience shows that there are two general kinds of accelerator operation-development and production. In development running, the operator is trying to push the state of the art [or, as (M. Hine 1979) points out, to repair the broken accelerator]. Change is the operator's goal, and also, to some extent, his technique. In production running, the operator is trying to maintain stability of operation. This operator's goal is integrated luminosity for elementary particle physics experiments. The existence of these two kinds of operations implies two kinds of operators and two sets of possibly overlapping control console requirements.

Although not strictly within the area of interest of this paper, the control center of ISABELLE will contain more than just the computer control system. Operators will need information from the beam monitoring and rf accelerating systems to run ISABELLE. Much of this information is of analogue nature, or is obtained from information processing equipment too expensive or bulky to duplicate at every control console. Experience with other accelerators indicates that it would be quite useful to have the low level rf system, the beam monitoring information processing equipment, and the control consoles in a single area or in contiguous rooms. The ISABELLE control room will be located close to the ring and cable runs between the control them and this instrumentation (pickup electrodes for longitudinal and transverse Schottky scans, beam current monitors, etc.) will be kept short.

The ISR Experience indicates that one operator (and therefore one console) should be dedicated to each of the storage rings. In addition the Engineer-in-Charge (EIC) or head operator should have a console. During production running, the EIC is charged with overseeing the operation of the colliding beam facility. The charge of the other two operators is clearly to keep the blue and yellow[1] rings stable. The SPS experience indicates that an additional console is needed for backup, and for the maintenance and development of new hardware and software for operations. "While a considerable amount of the program writing and modification can be carried out on the terminals of [a time-shared program development system], a console is necessary for testing and de-bugging if displays are involved, which is the case for most applications programs." (Crowly-Milling 1978). Thus, the conclusion is reached that four consoles are necessary.

Details of the control console design are not yet fixed. Following the example of the SPS, each control console will include a computer system. Operators will call accelerator application programs into execution which will, in turn, communicate with the various area computers and utilities for the purpose of information and control. Since the operator is interacting at the console with the accelerator, the applications programs can be quite large and complicated. Thus, the limitations of the addressing space set by sixteen bit memory may be too confining. Thirty-two bit computers are an obvious solution, and may very well be a cost effective solution.

Color video terminals are cost effective devices for allowing an operator to obtain an overview of the status of complicated systems. One can take advantage of the increased information per unit area available due to color (Crowley-Milling, 1978). This utility is available in a number of ways. Lists of numerical values can be color coded to show actual, set, transition, and out of limits values. System flow diagrams can be color coded with respect to both process flow and status. Of course, one can simply put more curves on the same "sheet" if the curves are drawn in different colors. This latter utility has only recently begun to be truly a cost effective possibility. One of the problems with color graphics has been the lack of a good, fast hard copy facility. Such units are now appearing in the market place, although they are still expensive. However, one can conceive of a single unit in the ISABELLE control room which is shared among the control consoles.

The control console includes a computer system. A terminal to communicate with that computer system is necessary. The combination of CRT and keyboard satisfies that need. Cursor control via a track ball has become almost ubiquitous in accelerator control console design. ISABELLE will follow that lead. Computer controlled touch panels allow one to have as many on-off "buttons" as needed and still use very little console space. One can call programs into execution and turn on and devices via the touch panel. The former utility appears to be one of the most used features of the SPS control consoles.

The use of a light pen might be considered. One of the common tasks an operator has during development running is to modify a time based function such as is used in a function generator controlling a power supply driving a ramp, for example. It is hard to do this using a track ball - drawing a two dimensional curve with a track ball calls for some dexterity. It would be easier to "draw" the curve by hand on the "face" of a crt. A light pen allows just such an operation.

Most of the above-mentioned features are needed in both types of accelerator operation. This is because the MMIF as interface to the control system has been emphasised more than the MMIF as interface to the accelerator. Thus, much of the above mentioned elements of the control consoles are conceived of as computer peripherals. When we begin to consider some of the higher bandwidth information, the differences between production and development running begin to appear. These differences are due, in part, to the differences in the type of operator who is controlling the accelerator during the alternative types of running. The development operator is the machine scientist (or rf engineer, cryogenics system designer, etc.). In trying to push the state of the art, he is trying to manipulate parameters in untried ways or combination. Thus, by hypothesis, well tried and de-bugged software is not available. Since the techniques being attempted are new, ways of presenting the data are also not available. Combinations of parameters imply a need for multiple graphics presentations. In addition, since the accelerator physicist is manipulating beam, he will need more analogue readout devices, such as fast oscilloscopes, than the production operator. These needs of the development operator imply that at least one of the consoles should have more elaborate fast analogue presentations than the others. In addition, there should be a language available to the console user for easily writing programs which has de-bugging and graphics

---

[1] Blue (yellow) is the designation for the clockwise (counterclockwise) ring of ISABELLE.

facilities. The interpretive language NODAL, used at the SPS, appears to be an obvious candidate to satisfy this particular need. BASIC may also be considered.

2. Data Base Facility and Program Library

In the largest sense of the term, a data base facility encompasses the program library. However, it appears to be common useage among accelerator users that data base refers to the parameters that describe the accelerator and the program library is the repository of programs used in operating the accelerator.

Any data base philosophy implemented in a distributed system is a compromise between two mutually exclusive needs-the need for a centralized data base and the need for a distributed data base. The advantage of a central data base is that the response to monitoring interrogations is quite fast. However the information obtained may be old. Maintaining a central data base in a distributed system necessitates a great deal of communication, which in turn puts high speed requirements on the computer communication technique. If one wants to take a "snapshot" of the state of the machine, it is only necessary to make a archival copy of the central data base or some portion thereof. This can occur quickly because there is no need to communicate to many distributed devices. The problem of software surveillance is much simplified, given a central data base. Surveillance programs can monitor the data base, rather than the devices. A major disadvantage of the central data base is that any parameter change must be communicated to it. Thus, parameter changes at remote locations must communicate with the central data base.

The SPS has implemented a distributed data base in its use of "data modules." A data module is a piece of software resident in the computer system closest to the actual physical device. All information concerning the device (addresses, operations, limits, set points, etc.) is contained in the data module. The set of all data modules thus contains the accelerator data base. The advantage of this approach is that the information obtained from the data base in "immediate." Parameter changes can be effected without needing to be communicated to a central facility. The disadvantages are that "snapshots" take much longer to do and surveillance puts more burden on the communications system.

The advantage of being able to effect a parameter change without having to communicate with a central facility appears to be the most weighty factor. A strong point in its favor is the fact that all the information concerning a device is located in one piece of software. This clearly makes the device much easier to de-bug in both the hardware and software aspects. A review of the goals of the ISABELLE control system shows that goals 2, 3, and 4 tend to put their weight on the side of the distributed data base approach. Thus, the conclusion is reached that ISABELLE will follow the SPS example and use a distributed data base.

Can some of the positive features of the central data base be retained? Yes, if it is decided to take advantage of ISABELLE's stability of operation. There can be two data bases in the ISABELLE computer system. The primary working copy will be the distributed data base in the distributed data modules. A secondary copy can be a central data base, resident in the "modelling computer." It will be necessary to update this secondary data base, which will mean heavier traffic on the computer network communications, which could conflict with the usage of the communications system for higher priority traffic. One could resolve this priority problem by in fact giving a lower priority in the message transfer system to messages which have to do with updating the central data base. One might incorporate this data base updating traffic into the message transfer self surveillance traffic (Crowley-Milling 1978). There is still the problem with the "aging" of the central data base, but ISABELLE must be stable for normal operation. Thus the fact that the data is the central data base may be "old" should not be critical. The immediate value of a parameter is always available in the data module in any case.

Should the program library be centralized or distributed? As far as data modules are concerned, an archival copy in a central location would be useful, but the backup copy should be located on disk or some kind of permanent media at computer where the data module is used. The difference between "backup" and "archival" is mostly the difference in the age of the data in the data module, the backup copy containing the more recent information. The actual programs themselves must be identical.

Should the applications program library be centralized or distributed? In general, applications programs are to be run in control console computers or other computer systems in the control center. These computer systems can be quite large, and disks or other storage media will probably be a part of each system. Thus, a distributed program library would appear to be the obvious choice. However, most operators will interact with the control consoles. If each console has its own program library, then the philosophy of having all consoles appear identical to the computer system is difficult to maintain. An operator may make a change to a program on his disk. This change would then have to be implemented on the disks of the other consoles. Unless such techniques can be implemented easily, the uniformity of the control consoles will soon be lost. A simpler approach would be to have a single program library (Crowley-Milling, 1975).

3. Analytic Computation Facility - The Modelling Computer

This facility specifically addresses the goal of being able to model ISABELLE in both the predictive and historical sense. Logically, it is a facility capable of performing complicated accelerator simulations in real time. Physically, it may be a combination of a link to some central scientific computing facility and a 32 bit (or larger) computer which is part of the ISABELLE control computer network. More complicated simulations could be done via the link at the central scientific facility, possibly in batch mode. The computer in the control computer network is a more dedicated facility. It would be used for doing simpler calculations (but not simple calculations - thus the need for 32 bits). Since the model for the accelerator exists here, this is a natural place to have the data base available - allowing convenient comparison between theory and reality. This feature is clearly of use for the purposes of surveillance also.

The phenomenological models of the accelerator will take some time and actual operational experience before they can become reliable aids in the running of ISABELLE. The impact of a reliable predictive model which would be interrogated whenever substansive changes are made must be investigated carefully. Clearly, a model which is interrogated whenever any change, no matter how small is made, possesses most of the disadvantages of the central data base facility, plus its own disadvantage of imposing some delay due to a necessary calculation or table lookup to see if the requested operation is acceptable.

4. Program Development Facility

Control consoles are too precious a commodity to be used as common terminals in the writing of programs and reports. In addition, many programs, such as surveillance programs, systems programs, data modules, etc., have little or no need of the graphics and other utilities of a control console. A large computer with a time shared operating system to support a number of terminals is an obvious candidate for a cost effective facility which will support a great deal of program development. In addition, it could act as a backup for the modelling and data base facility to increase the over all availability of these critical control system utilities.

Since most of the actual programs used in the ISABELLE control system will be developed on this facility, this is an appropriate place to discuss what languages will be used in the ISABELLE control system.

The field of computer programming languages is very rich, and getting richer. Philosophies of language use on computer systems run from assembly language only systems used in dedicated or embedded systems (such as device controllers) to the anarchy of large computing facilities with the capability of supporting almost any conceivable computer language. What should be ISABELLE's philosophy? The projected lifetime of ISABELLE and its control system is between two and three decades. The necessity that programs written in the first decade of ISABELLE's existance be understandable and modifiable requires that they be written in a language which will continue to be supported by the ISABELLE control system during ISABELLE's lifetime. In order that goal 4 be achieved, the languages chosen should be useful and popular at least among the ISABELLE control system users. Many of the design personnel of ISABELLE are acquainted with FORTRAN and BASIC. Both of these languages are standardized and well supported by the industry. The language standards which are or will be available include real-time extensions for process control and synchronization mechanisms for operation in a multi-task environment. Both these languages appear to satisfy the requirements of popularity, utility, and lifetime needed for ISABELLE. It can be argued that NODAL may be of greater utility than BASIC because of its greater graphics and string manipulation utilities. Certainly its development in an accelerator environment makes it a strong contender for replacing BASIC at ISABELLE.

In addition to the aforementioned languages, assembly language will of course be available to the knowledgeable user. However, the field of computer science has become very enamoured of procedural languages - specifically PASCAL-which afford more transportability, meaning that PASCAL is supported by different computer manufacturers, than assembly language and more utility for systems programming problems than FORTRAN, BASIC, or COBOL. If the present enthusiasm of the computer manufacturerers to support PASCAL continues, making a decision for or against its use may become moot. Even though it may not be used by control systems users, PASCAL seems an obvious choice as a more efficient alternative (because it is a high level language) to assembly language for systems programming problems.

Thus the languages proposed for use in the ISABELLE control system are FORTRAN, BASIC (NODAL?), PASCAL, and Assembly Language. In such an environment, two additional points must be made. An efficient, easily used word processing facility for software documentation must be available. Perhaps a managerial policy that no program will be allowed to reside on the operations disk for more than two weeks unlesa a documention file is available might be enforced. The second point is that subroutines written in one language should be callable by another. For compiled languages, and for subroutines called by an interpreter, such facilities are already available. However BASIC and NODAL are interpretive, and facilities to call an interpreted BASIC subroutine from a FORTRAN, PASCAL, or Assembly Language Program do not yet exist - to the author's knowledge. A simple solution is to have both an interpreter and a compiler for BASIC. Such facilities are available from at least one computer manufacturer. A compiler doesn't yet exist for NODAL, but could be written if deemed necessary.

5. Alarms and Access

As the design of the ISABELLE Control system becomes more developed it may be that the alarms and access functions become more separated. For now, it appears that the functions are complementary enough to be connected together in a single computer system.

The alarms function is intimately connected with the surveillance function. One surveys to find parameters which are out of limits, and then broadcasts an alarm. In the environment in which a central data base is available, much of the surveillance of accelerator parameters can occur via that data base. Thus a parameter surveillance program could run in the central data base facility, and alarm states can be communicated to the alarms computer. Some parameters will possess time constants such that surveillance in the central data base is deemed too slow. In those cases, surveillance can occur in the data module, or even in the device itself. In this latter case, an interrupt or service request facility must be available so that the device can gain the attention of the local computer to inform it of the alarm condition. The alternative to this interrupt facility is a polling scheme where the local computer polls all devices asking if they have an alarm condition. In any case, it is clear that alarms can come from various sources to the alarms computer. In addition to parameter surveillance of the accelerator, a similar function must be performed for the central computer network. The communications links must be surveyed. There are two candidates for this function. The obvious one is of course that computer which is dedicated to the communications system. Another possibility is the alarms computer because, like the communications or message handling computer, it has systems wide responsibilities and is naturally in communication with all computers of the network.

Having received notification of the alarm state, the alarms computer will then transfer the appropriate alarm message to each control console, which will have a video terminal dedicated to alarms. Depending upon the priority of the alarm, the operator may decide to follow different courses of action which may range from mere acknowledgement to emergency beam abort. In addition to the alarms terminal at each control console, the alarms and access computer will have its own console in the control room. Alarms messages will also be recorded on this terminal also.

During beam off periods, access in ISABELLE will be controlled from the access and alarms console in the main control room. The access console will have video monitors which can be switched to any of the TV cameras covering the various points of personnel entrance and exit. The operator can thereby monitor access.

An important aspect of the access system is that it is most used during maintenance periods. However, accelerator maintenance includes control system maintenance, which may imply downtime for the network communications system. Thus the access system needs to have communications to the central access console which are independent of the control computer network. (The alternatives of a guard at each entrance/exit or uncontrolled entrance/exit are not considered viable for economic or operational reasons.)

IV. CONTROL COMPUTER NETWORK

The distributed systems approach upon which the ISABELLE Control System design is based uses the concept of a widely distributed set of functions carried out by small and relatively inexpensive computer systems (Purdue University, 1977; Crowley-Milling, 1975; Dimmler, 1978). These distributed parts are unified into a system by means of the control computer network which supplies the means of communication, and the various software and hardware utilities to support parallel processing and shared resources. A computer network in an environment such as ISABELLE is considered to be a local network. A local network is local in two senses:

1. It is generally owned by a single organization.

2. It is geographically local; i.e. distances are on the order of a few miles (Thurber, 1979).

Both of these attributes of a local network have positive aspects that reduce the complexity and, therefore, the cost of the network implementation - especially in ISABELLE's case. Since ISABELLE is a new single organization, a single type of computer hardware/software can be managerially imposed upon the solution. Thus the implementation of specific protocols and interfaces can be kept to a minimum. The presence of different operating systems or instruction sets could necessitate multiple software interface implementations, with a concomitant increase in the cost of implementation and maintenance. Geographic locality can mean that the error rates and delay of communication lines are drastically improved over those used for non-local distances. Thus specifications on delay and throughput can be met with less expensive technology than would be needed for similar performance in non-local networks.

6. Computers

Figure 2 shows a block diagram of the control computer network with the functions ascribed to the different nodes (computer systems) of the network. The architecture shown is referred to as a star, where the nodes at the points of the star communicate through a central communication system.

That portion of the nodes which are part of the control center have already been discussed. The assignment of the other twelve nodes has been based on the philosophy that there are functions of sufficient complexity and isolation that a computer system can be effective in the autonomous control of that function. Thus, for example, the cryogenic system or vacuum system can be operated independent of the power supply system, for the most part. Some of the functional assignments of the different nodes are obvious - rf, cryogenics, power supplies, and vacuum. The separation of injection from ejection is less obvious, but is made rational on the basis of the criticality of the ejection function from the point of view of personnel and accelerator safty. Although the computer system itself will not be involved in the triggering of the extraction sequence, the monitoring of that sequence before and during the event is of vital importance for the historical phenomenological model. In addition, the injection computer assignment is still under discussion. It may be that that computer is the AGS control computer itself (presently a PDP10). Such a design decision clearly subtracts from one of the advantages of a local computer network. A possible solution is to replace the PDP10, but this may not be cost effective. Another solution is to use the concept of the gateway (or interface) computer. This is a computer (of the same type used in the rest of the ISABELLE network) that would be placed between the ISABELLE computer network and the AGS PDP10. Thus, as far as the ISABELLE network is concerned, all the computers are identical. The gateway computer then performs the necessary translations between the AGS control system and the ISABELLE control system. While such a solution does not abrogate the basic problem of needing two software interfaces, it does localize the translation function into a single computer system which is independent of the AGS and ISA control functions. Such a gateway computer is clearly a minimal cost hardware configuration.

The assignment of a computer system to each sextant bears more investigation, and is intimately connected with the design philosophy of the process interface communications topology. For example, does the ISABELLE vacuum computer communicate with vacuum devices over its own communication link? Or does it communicate with the appropriate sextant computer which then communicates with the vacuum device via a link shared by all ISABELLE subsystems? In this latter case, operation of the vacuum system, qua system, is not possible unless the network communication system is functioning. This fact then impacts the availability specification of the network communications. Experience with the SPS shows that the fraction of accelerator downtime attributable to the control system can be held to the 1% or less level (Crowley-Milling, 1978).

What kind of computers should be used? Considering the present marketplace, this question reduces to the question 16 or 32 bits? The answer is 32 bits because the complexity of ISABELLE requires large application programs which need the addressing capabilities of the 32 bit machines. While it is true that large programs can be run on 16 bit machines using overlay techniques, the efficient use of such techniques requires more sophistication on the part of the user programmer than is consistent with goal number 4. It is possible that a mixture of 16 bit and 32 bit machines may be used. One computer vendor makes a 32 bit machine which, in addition to its own instruction set, executes the instruction set of a 16 bit computer manufactured by the same company. This would allow ISABELLE to assign 32 bit machines to functions where the large address space is needed and 16 bit machines where it is not. On the other hand, the 32 bit marketplace is an active one, and a cost effective solution of using 32 bit machines at all nodes of the network may very well be possible.

7. Network Techniques

What kind of local computer network (LCN) techniques will support the star-like configuration shown in Fig. 2.? Two techniques appear feasible - the message handling computer and the contention system. The SPS control system uses a message handling computer. Examples of contention system are Ethernet and Hyperchannel.

The message handling computer that the SPS uses is a polling, store and forward technique. Each node on the star is polled in succession to see if it has a message. If node A replies yes, the message is received in the central computer and retransmitted to the intended receiver (node B; see Fig. 3). Another possibility along similar lines is the circuit switch. In this case node A tells the message handling computer that it has a message for node B. The message handling computer then sets up a physical circuit between A and B over which the message is sent with no pause at the MHC. In an LCN environment, there is an important difference between these two. Because of the necessity to store and retransmit, the former technique becomes slower than the circuit switch for long messages. For short messages however, the store and forward technique may be faster - due to the necessary overhead to set up the circuit of the latter technique.

The contention technique is presently receiving a great deal of attention in the LCN community. (See Fig. 4). In this method, all nodes are connected to a common high bandwidth coaxial cable. A station wishing to transmit senses whether or not there is a carrier signal on the coaxial cable. If there is, a message is already on the line, and the station waits until the line is quiet. If (or when) the line is quiet, the station then transmits its message. All message include error checking words to detect a corrupted message - as would occur if two or more messages collide with one another. If collision occurs, the message is retransmitted. This technique possesses good characteristics for both short and long messages, as well as the feature that broadcasting (one sender, multiple receivers) is obviously included. Broadcasting is difficult to incorporate into either of the message handling computer techniques. Broadcasting is clearly a useful procedure for program synchronization.

In terms of performance objectives, there are two measures of the speed of a communications scheme - delay and throughput (McQuillan, 1978). Delay is usually measured in terms of average response time, throughput in terms of the peak traffic level supported. In a control environment, which must *respond* to real accelerator events, it is delay that is more important than throughput. The difference between the two concepts is commonly described by the difference between a satellite link (delay = 0.25 sec, throughput = 1500 kbs) and a common voice grade telephone circuit (delay = 0.01 sec, throughput = 2.4 kbs). For the former case, the time to send a 100 bits is 0.25 sec, and, for 1000000 bits, 0.92 sec. In the voice grade circuit case, the corresponding times are 0.05 seconds and 417 seconds.

In the case of ISABELLE, what are the time constants of most interest? In terms of response time, the fastest event that the control system might be expected to respond to is a quenching magnet, whose typical time constant is of the order of 100 milliseconds. A more significant time is the period magnet takes to loose 1% of its field - which is approximately 5-10 milliseconds. It could be of great utility to be able to transmit a message through the network before the magnetic field has significantly changed. This would be a short message of the highest priority, and would set the delay specification. (This message would be in addition to the triggering of emergency quench procedures, which would be handled in a system separate from the control computer network). Thus a delay of 5 ms for the sending of the highest priorty message from one computer node to another would be set by this analysis.

What is a reasonable specification on throughput which would be set by accelerator operation? The SPS has done an in depth analysis of traffic in their computer network. The main traffic is among computers which, in the terms of this paper, are in the control center. In other words, traffic to and from the accelerator is not a major component of the total network traffic. However, throughput is measured in terms of *peak* traffic level supported, not average. In the case of ISABELLE the peak traffic induced by accelarator operations would be during the transmission of data acquired during an extraction event. Let it be assumed that there are approximately 2000 circular buffers of 1000 bytes each which are constantly acquiring magnet coil data. This data is only read out after an extraction event has occured. There are thus 2000 monitor point x 1000 bytes x 8 bits/byte ≈ 20 x 10^6 bits of information. To read this event onto a disk in one minute is an upper limit to the length of time one would want to wait. 20 x 10^6 bits in 50 sec means a throughput of 400 kbs.

Measured results on the SPS message handling computer show a delay of 4 milliseconds and a throughput of 688 kbs (using a technology which will be 15 years old when ISABELLE turns on) (Crowley-Milling, 1978; Altaber, 1978). The ISABELLE requirements appear to be conservative, in terms of what the technology has already achieved. A facility for monitoring the performance of the network communication system must be available. Such a facility would include diagnostic and debugging tools. In the case of a system based on a message handling computer, such a facility may be present in the message handling computer itself. In the case of a contention system a computer dedicated to monitoring and diagnosis of communications traffic may be necessary to fulfill this requirement.

8. The Process Interface

The process interface is a mixture of hardware and software. Specifically, the process interface is the device dependent software in the local computer, the communications hardware and software which connects that computer to the process device, and it is the data and command structures in the device itself.

It is important to point out that a device is not just the control electronics which is connected to an equipment; a device is the equipment *and* its controls. Thus, device design is not necessarily the responsibility of the Controls Group. That responsibility resides in the group building the equipment.

It is the process (or device) interface that will be impacted most by the distributed approach of the ISABELLE Control System. The advent of microprocessors and other LSI (large scale integration) techniques allows the device designer the option of making intelligent instruments (or devices). It is this development that has allowed the possibility of achieving a control system whose goal is to do only set point control. All DDC (Direct Digital Control), e.g. digital feedback, can be off-loaded from the control computer network to the device. Control algorithms will be resident in the hardware and software of the actual device itself. The only information necessary to be transmitted to the devices will be high-level commands (off, on, standby, receive or transmit function table, etc.) and data.

9. Device Design Philosophy

Many of the devices of ISABELLE will be contained in the equipment areas of the ring. These areas (see Fig. 5.) are not accessible during operation. In order that ISABELLE operate reliably, equipment in these areas must be designed to operate without human intervention. It is just in this kind of environment that the distributed approach can be effective - if the appropriate design techniques are followed. Devices should be self contained logically and not require the constant intervention of the computer system. Such intervention is possible, but it puts great demands on the throughput of the process data highway (the communications system between the computer network and the device). Thus, the keywords concerning device design in the distributed ISABELLE environment are self-containment and reliability. To achieve these goals, the philosophy of device design can be summarized as follows:

ISABELLE Device Design Philosophy

1. No device should require the local computer to be dedicated to a closed loop. If such a closed digital loop is necessary, it is the responsibility of the device designer to incorporate it in the device.

2. Only data and commands can be transmitted between the control computer and the device. Specifically, the device designer is not allowed the option of having programs down-line loaded into his device.

3. All devices will communicate with the computer network via a standard communication using standard protocols. Command and data structures used in devices will also conform to control system standards.

The last point, albeit specifically addressed to the device side of the process interface, also applies, in part, to the computer side. The data module is the combination of software driver subroutine and data base which contains all the information about the device known to the control computer network (limits, set points, commands, etc.). Any program seeking to control or monitor a device does so via the data module. Since this facility will be used by numerous individuals, it too must conform to control system standards with respect to data formats, command structure, and documentation.

10. Process Data Highway

The process data highway is the means of communication between the control computer network and the actual device or process. Previous accelerator implementations of the process data highway are Datacon (AGS), SEDAC (PETRA), Serial CAMAC (ISR), and the MPX (SPS). Of the four, only serial CAMAC has been used elsewhere. None of the other three can be considered more than single-site solutions to the more general problem of distributed process interfacing. Serial CAMAC has been used elsewhere, but suffers from the detriments of being expensive (Rausch, 1976), lacking in reliability (in the process environment) due to its limited error checking capabilities and poor connector design, and being crate-oriented. This last detriment means that it is somewhat incompatible with the philosophy of having intelligence located in the controlled device. CAMAC itself is a mechanical crate and bus standard. Serial CAMAC is a means to communicate with a series of CAMAC crates. It is not a means to communicate with a series of autonomous intelligent devices.

As in the case of the local computer network techniques, no design decision has yet been made. A preliminary report on the subject is due in January of 1980. What are the functional requirements of the process data highway? A working group (WG6) of the Technical Committee on Industrial-Process Measurement and Control (TC65) of the International Electro-Technical Commission (IEC) has begun to circulate draft versions of the functional requirements of PROWAY - a process data highway for distributed process control systems. This is a rather extensive document, (British Electrotechnical Committee, 1979), too lengthy to reproduce here. Many of the requirements and definitions are applicable to the ISABELLE environment. This committee defines the optimum characteristics of a process data highway to be:

"1. Event driven communication which allows real time response to events.

2. Very high availability.

3. Very high data integrity.

4. Proper operation in the presence of electromagnetic interference and differences in earth potential and,

5. Dedicated intra-plant transmission lines."

Typical delays mentioned in the document are less than two milliseconds and throughputs mentioned are in the range of 30 to 1000 kbs. ISABELLE personnel are actively following the development of PROWAY.

What might be the architecture or topology of the process interface? As mentioned earlier, many devices in ISABELLE are contained in the equipment areas of the magnet enclosure. Other locations are the service building, equipment areas at each intersection region, the rf building, the cryogenic compressor building, etc. The process data highway must connect to the devices in these areas and then transmit information to a fewer number of locations where connection to the local computer will take place. An obvious topology would be to connect all the devices within a relatively short distance (~ 100 ft) to a data concentrator and then transmit over longer distances to a connection to the local computer (see Fig. 6). (In fact, this is the model used in the PROWAY functional specification). Such a technique decreases dramatically the number of wires (cables) that must be stretched over long distances.

Connection of the equipment alcoves to the local computer may occur in a number of ways, but two obvious candidates are shown in Figs. 7 and 8. The hierarchical star of Fig. 7 follows the example of the SPS configuration. All the equipment on a sextant connects to one of the sextant local computers. A problem with this configuration is that one cannot communicate with all vacuum or cryogenic devices unless the computer network communication system is functioning. However, due to the fact that each sextant can be communicated with in parallel with the others, the total throughput of information is six times that of a single computer collecting data from the entire ring.

Figure 8 shows a loop or broadcast connection in which a single dedicated system computer (cryogenics, vacuum) communicates over a single link to all the devices of that system. It is quite common that such links have redundant paths for reliability considerations. Such a topology solves the problem mentioned in the preceding paragraph, but at the expense of decreased throughput.

It may be that a combination of the two solutions is the best technical choice. The cryogenic and vacuum systems, because of their operation, require that they be functional for up to 2-3 weeks before accelerator operation. Systems such as beam instrumentation and power supplies are much more tightly coupled to the presence of beam when operation as an accelerator system is considered. The rf system is very local in any case (the low level rf system and power supplies are in the service building, the high level rf system is in the rf building). In addition, the beam instrumentation can have much higher throughput and lower delay requirements than the cryogenic and vacuum systems. Thus, the hierarchical star à la SPS may be the best technical choice for the beam instrumentation, power supplies, and magnet monitoring system.

What technique should be used to connect devices in an alcove to the concentrator? One possible candidate is IEEE-488. This instrumentation-oriented standard was designed to connect devices over relatively short distances to a computer. This design goal of IEEE-488 is very close to the kind of function needed in an alcove. In addition, the standard comes impressively documented and well supported by the LSI industry. Our experience with IEEE-488 indicates that it is much more compatible with microprocessor designs than CAMAC or DATACON. In addition, it has a limited interrupt facility (service request) which is necessary in an environment where the control computer network is not constantly monitoring a device. The connection between an intelligent device and interrupt capability is so strong that the IEC, in the glossary section of PROWAY document defines an intelligent station as one "which includes application units [devices] capable of initiating and controlling message transactions through a Data Highway." In fact, IEEE-488 interfaces will exist as part of the ISABELLE control system because of its acceptance by the instrumentation industry. However, a commitment to use IEEE-488 as a process interface in addition to using it for various industry supplied instrumentation (digital voltmeters, sweep generators, network analyzers, etc.) must await a more detailed analysis of the cost effectiveness of the standard in the ISABELLE environment.

11. The Timing System

An accelerator is a single device comprising many systems, some of which must be synchronized or time ordered (Crowley-Milling, 1975). Specifically, during the acceleration period, the power supplies must be ramped in synchronism with one another and with the accelerating rf system. In addition, measurements of beam location from different orbit monitors must be synchronized, or at least time ordered. If a superconducting magnet reverts to the normal state, measurements of beam location, extraction, temperature rise, etc. must be time ordered. A clock is a necessity.

The rate of the clock lies somewhere between the rate at which protons circle ISABELLE (10 microseconds per revolution) and the characteristic time of a quenching magnet (~ 100 milliseconds). The present philosophy is to use a 10 kilohertz clock which is transmitted to each equipment area around the machine. Within the equipment areas, a local 10 kilohertz clock is locked to the master clock and transmitted over the ac power system using FSK (Frequency Shift Keying) techniques.

12. Review of System Goals

How does the system proposed meet the goals outlined at the beginning of this paper?

1. "All ISABELLE system will be operated from a single control center." A single control center for ISABELLE will exist. All systems connected to the control computer network will be operable from that control center. The existence of a central backup data base, program development and modelling facilities and extensive control consoles will make operation from that center efficient and attractive.

2. "In the case of the loss of all or part of the control computer network, accelerator systems will continue to operate at the most recent setpoint." The use of intelligent, self-contained devices and the policy that

the control computer network will not engage in direct digital control gives the accelerator an effective buffer in the case of a network failure.

3. "The hardware and software interface of devices to the ISABELLE control system should be implemented in standard ways." The enforcement of software and hardware standards and the checks and balances available by forcing all accesses to devices to go through the common data module is a direct and effective response to this goal.

4. "The generation of programs and the testing of new devices, should be routinely done by control system users." The use of common well known languages such as FORTRAN and BASIC will make programming very approachable by control system users. The use of computers with large addressing space will not make it necessary for the users to understand the complexities of overlaying and other esoterica for the implementation of large programs. The use of standard data modules, synchronization and data base procedures will allow the users a uniform approach to device control.

5. "The control system should be capable of phenomenologically modelling the state of ISABELLE - both in a predictive and historical sense." A powerful modelling facility exits for this specific goal.

V. ACKNOWLEDGEMENTS

Readers familar with accelerator control systems will recognize the debt owed to M. C. Crowley-Milling and his colleagues of the SPS. Colleagues associated with accelerators at CERN, FNAL, SLAC, and LASL have also contributed. The International Purdue Workshop on Industrial Computer Systems has been a fruitful contact with colleagues in the industrial community. Naturally, colleagues at BNL have contributed ideas, clarifications, and much appreciated constructive criticism. Too numerous to name, they are associated with the Accelerator, Physics, and Applied Math Departments, the Instrumentation Division, and, of course, the ISABELLE Project. Robert Frankel of the ISABELLE Project has been an invaluable resource in the development of the concepts described in this paper.

VI. REFERENCES

British Electrotechnical Committee (1979). Technical Committee No. 65, Intern. Electrotechnical Commission, Draft - Process Data Highway (PROWAY) for Distributed Process Control Systems, IEC 65A (Secretariat), London, Great Britain.

Crowley-Milling, M. C., (1975). The Design of the Control System for the SPS, CERN Report, CERN 75-20, p. 9.

Crowley-Milling, M. C., (1978). Experience with the Control System for the System for the SPS, CERN Report, CERN 78-09, p. 17.

Crowley-Milling, M. C. (1978). *Experience with the Control System for the SPS*, pp. 24-26.

Crowley-Milling, M. C. (1978). op. cit. p. 5; Altaber, J., Jeanneret, J-P, (1978). Measurements on the Control Network of the SPS, CERN Rept. CERN/SPS/ACC/78-1, Translated by F. Beck, CERN, Geneva, Switzerland (1978).

Crowley-Milling, M. C., (1978). op. cit. pp. 5-6.

Dimmler, D. G. (1975). ICAM - The ISABELLE Control and Monitoring System, Global Overview, BNL Report, CRISP 75-5; Lamport, L. (1978). Time, Clocks, and the Ordering of Events in a Distributed System, Communications ACM, *21*, pp. 558-565.

Dimmler, D. G., (1978). Computer Networks in Future Accelerator Control Systems, IEEE Trans. Nucl. Sci. *NS-25*, No. 2, pp. 974-988.

Hine, M., (1979). Private communication.

MMIF Committee, (1978). Guidelines for the Design of Man/Machine Interfaces for Process Control, Intern. Purdue Workshop on Industrial Control Systems, Purdue University, 2nd Rev. p. vii cf. "and what is the use of a book" thought Alice, "without pictures or conversations?" Lewis Carrol, Alice in Wonderland (1865) quoted by F. Beck, The Design and Construction of a Control Center for the CERN SPS Accelerator, CERN Report. SPS-CO/76-1 (1976).

Purdue University, Project Staff, (1977). Tasks and Functional Specifications of the Steel Plant Heierarchy Control System, Report #98, Purdue Laboratory for Applied Industrial Control, pp. 2-3 to 2-5.

Rausch, B., (1976). Cost Comparison Between Serial CAMAC and SPS-MPX System for Control Applications, CERN Rept. SPS-CO/Int/CC/76-6.

Thurber, K. J., Freeman, H. A. (1979). A Bibliography of Local Computer Network Architecture, Sperry Univac, St. Paul, Minnesota.

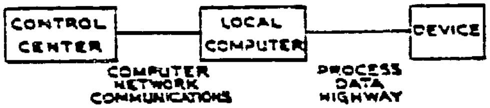

Fig. 1. Conceptual Computer Control System.

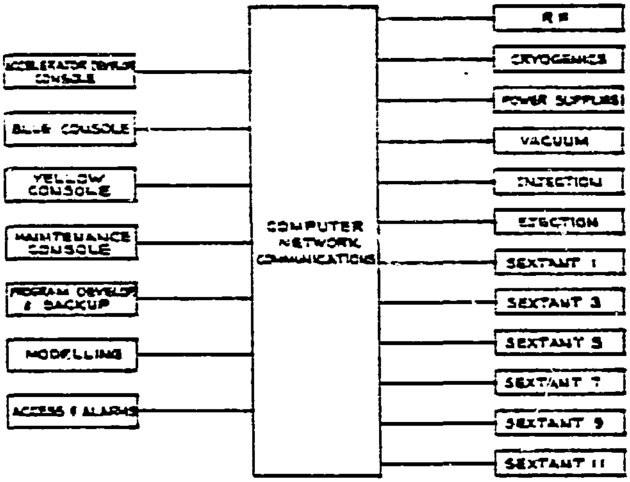

Fig. 2. Computer System Layout.

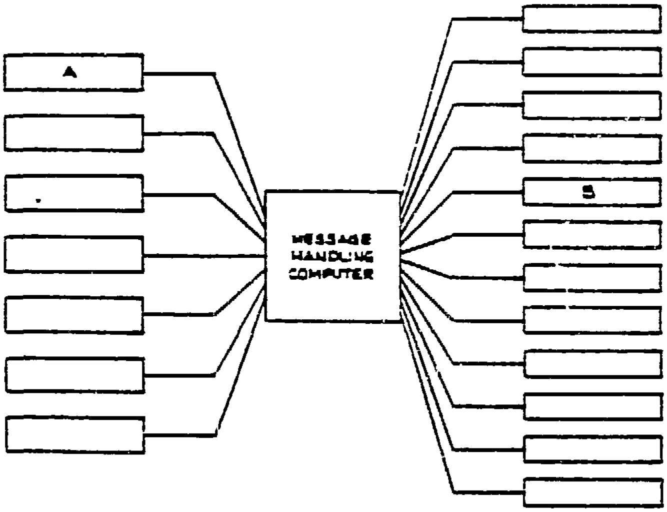

Fig. 3. Local Computer Network Based on a Message Handling Computer.

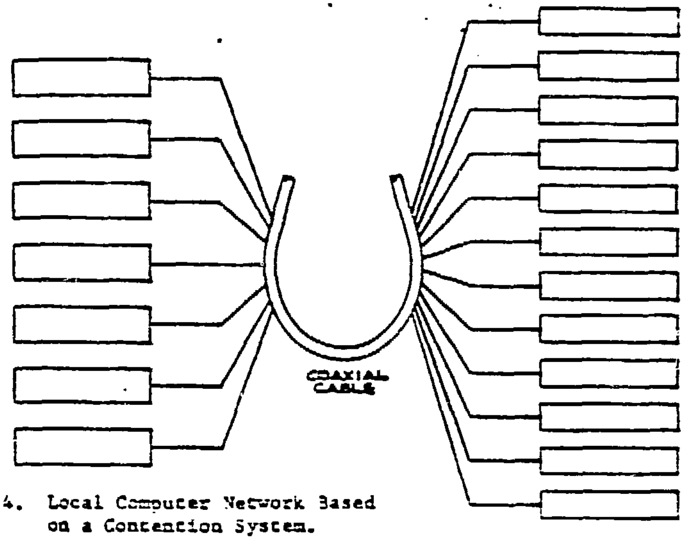

Fig. 4. Local Computer Network Based on a Contention System.

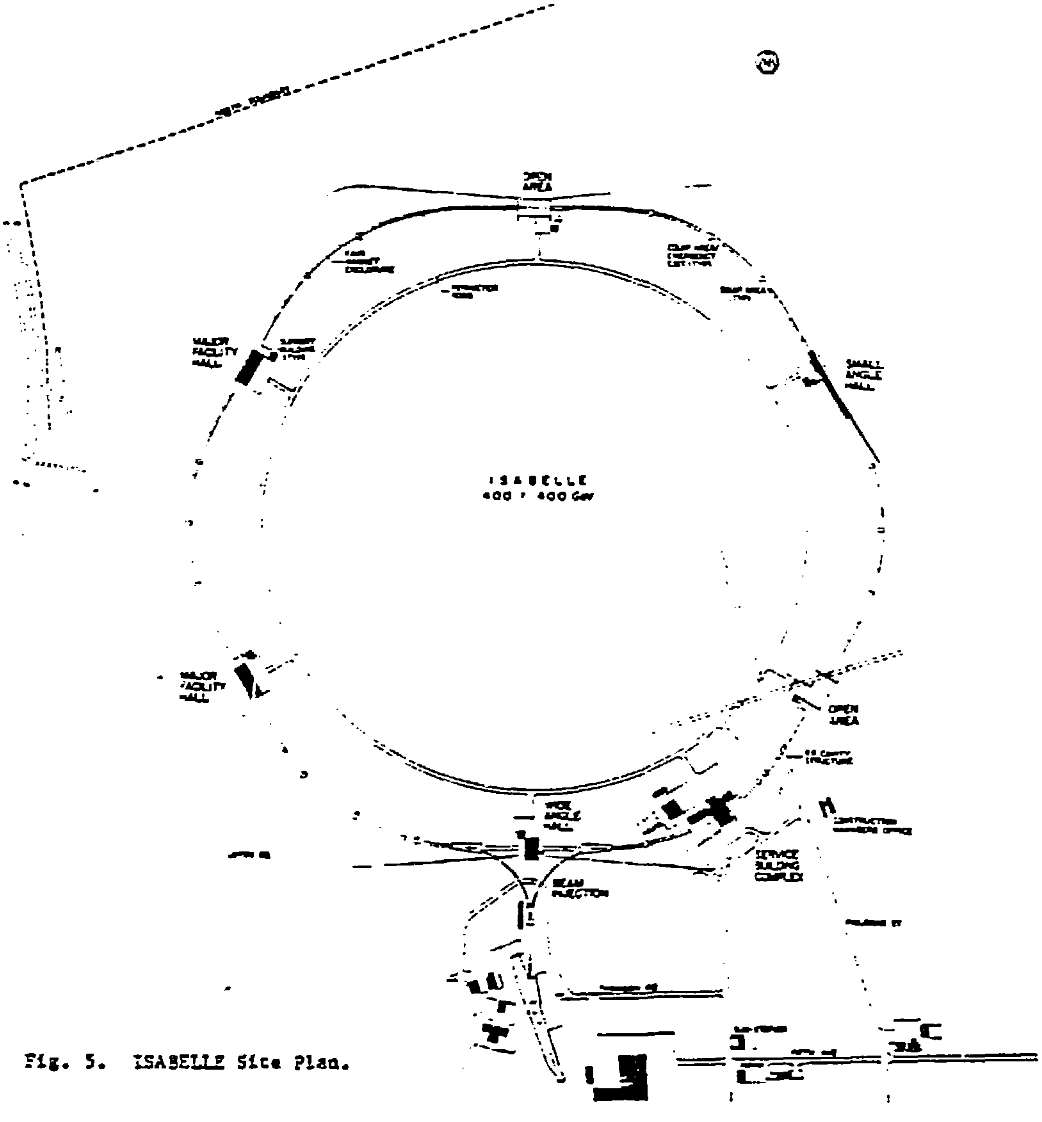

Fig. 5. ISABELLE Site Plan.

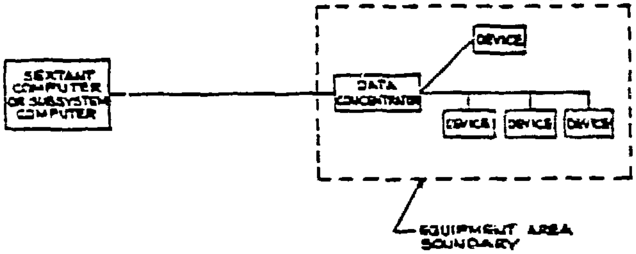

Fig. 6. Conceptual Diagram of Communication between a Local Computer and Devices in an Equipment Area.

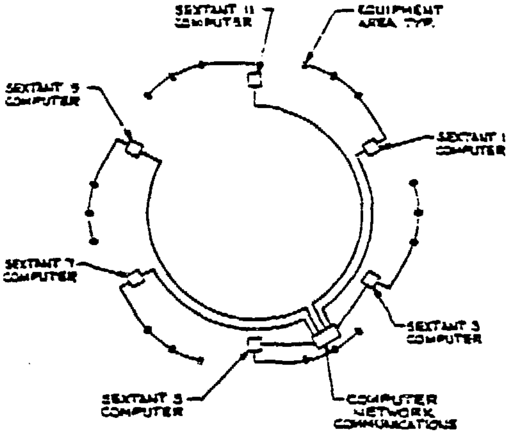

Fig. 7. Star topology for the Process Data Highway.

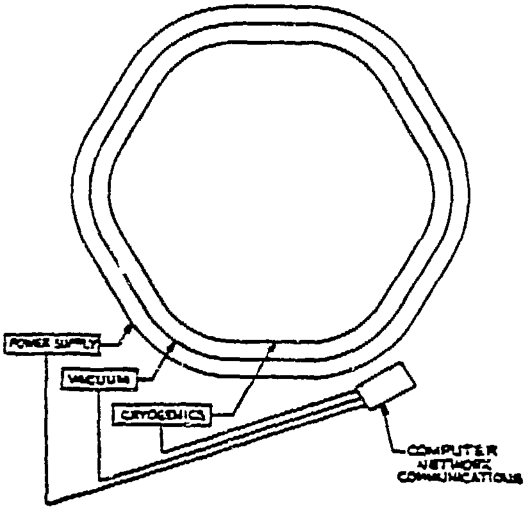

Fig. 8. Loop topology for the Process Data Highway.