SURVEY AND EVALUATION OF DISTRIBUTED COMPUTER CONTROL FOR THE WATER TREATMENT PLANT OF A THERMAL POWER STATION

Ephraim Nbuso Dlamini

A project report submitted to the Faculty of Engineering, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master Science in Engineering.

Johannesburg 1985

DECLARATION

I declare that this project report is my own, unaided work. It is being submitted in partial fulfilment of the requirements for the degree of Master of Science in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

EPHRAIM MBUSO DLAMINI

(Name of candidate)

8th day of AUGUST, 19 85.

ABSTRACT

This project report studies and evaluates distributed computer control systems (DCCS) as applied to the water treatment plant of a thermal power station.

The main problems encountered in such control systems are those of maintainability, reliability, data storage and communication and system architecture. The application of DCCS partly solves some of these problems.

In this project report these problems are analysed and the following possible solutions offered:

Maintainability	: Smallest Repaireable Units (SRU's) with special features are discussed.
Reliability	: Components with a long lifetime, and the use of back-up are discussed.
Data Storage	: A distributed data base with special features is discussed.
Communication System	: A distributed communication system with special protocols is discussed with reference to the problem of collisions and many other communication problems.
Architecture	: The system architecture is totally distributed, that is, there are no centralized hardware or software components.

In chapter 1 of this project report, the problems encountered in process control system design for the water treatment plant of a thermal power station are discussed. A survey of DCCS technology and building blocks to be used, provides the framework for the theoretical development later in chapters 2 and 3 in the project report. The case study DCCS requirements are then examined in chapter 4; and chapter 5 gives a discussion of the recommendations and concluding remarks.

PREFACE

Since 1982 the author has been involved in the design of the control systems for the water treatment plant of two of Eskom's thermal power stations.

Such control systems are divided roughly into:

a) Analog control which was single input and single output (SISO) variable type of control system.

b) Binary control which was also SISO

c) Sequence control which was in software, also SISO.

During the course of this work, it was found that it was not the way the design was implemented in hardware/software that was a problem; but the problem was rather the basic design principle which called for SISO type control system structures. This not only led to a lot of control equipment but also the cabling was extensive, that is, large equipment/control rooms and large cable racks were required and thus the expenditure rocketed.

This is what inspired the author to propose DCCS for these plants in order to get rid of the large centralized equipment room, to minimize the size of the control room and only [illegible] one bus cable connecting a number of self-contained local controllers (or automation islands) at the "same level".

The field of Distributed Computer Control Systems is an immense one and it is not possible to explore all the aspects in a project report such as this, but it is hoped, that the guidelines given in this project report will be used in this type of control system design in the future.

PREFACE

Since 1982 the author has been involved in the design of the control systems for the water treatment plant of two of Eskom's thermal power stations.

Such control systems are divided roughly into:

a) Analog control which was single input and single output (SISO) variable type of control system.

b) Binary control which was also SISO

c) Sequence control which was in software, also SISO.

During the course of this work, it was found that it was not the way the design was implemented in hardware/software that was a problem; but the problem was rather the basic design principle which called for SISO type control system structures. This not only led to a lot of control equipment but also the cabling was extensive, that is, large equipment/control rooms and large cable racks were required and thus the expenditure rocketed.

This is what inspired the author to propose DCCS for these plants in order to get rid of the large centralized equipment room, to minimize the size of the control room and only have one bus cable connectiong a number of self-contained local controllers (or automation islands) at the "same level".

The field of Distributed Computer Control Systems is an immense one and it is not possible to explore all the aspects in a project report such as this, but it is hoped, that the guidelines given in this project report will be used in this type of control system design in the future.

ACKNOWLEDGEMENTS

The author wishes to sincerely thank the many people who have contributed in various ways to the work reported in this project report. A full list of all those involved would be far too long to include here, but particular mention must be made of the following:

- My supervisor, Professor IM Macleod, professor of Control Systems, University of the Witwatersrand, for creating the necessary opportunities and for his guidance and enthusiastic support.

- J Mthimkhulu and P Tshauke for proof-reading the final manuscript.

- Miss C Merafe, for typing the final manuscript.

The author also wishes to acknowledge the provision of financial support by SIEMENS Limited. The research reported in this project report would not have been possible without this support.

CONTENTS

LIST OF ILLUSTRATIONS

List of Tables

GLOSSARY

DEFINITIONS

COMMUNICATION

ENTITY - It can be considered a subsystem, which again consists of subsystems, etc.

EVENT INFORMATION - It is concerned with a "happening" at a point in time, i.e. the occurance of an event.

FIBRE OPTICS - A transmission medium that consists of hairthin strands of highly transparent glass fibre. Data signals are converted to pulses of light for transmission over the fibre and back to electrical pulses at the distant end.

LOCAL AREA NETWORK TERMINOLOGY:

BASEBAND: A means of combining data signals by pulsing directly on the transmission line.

BROADBAND: A means of combining data, voice and video signals by modulating to different radio frequencies.

COAXIAL CABLE: A tansmission medium that consists of a single centre conductor surrounded by an insulator and shield.

PROTOCOL - It is a set of rules for the communication between entities describing:
- the syntax, i.e. the structure of the commands and responses.

- the semantics, i.e. the meaning of the commands and responses.

- timing, i.e. the specification of the time order events and the duration of intervals.

TOPOLOGY: The configuration of terminals in a network. The common networks use star, ring and bus topologies.

TRANSMISSION LINE: The physical medium for connection devices in a LAN. Coaxial cable, paired cable and fibre optics are most commonly used.

GENERAL

IMPURE WATER = non-demineralized or non-clarified water.

INSTRUMENTATION - It is the [illegible] between the control system and the production process.

INTERFACE - It is the boundary between a system and its environment.

PRODUCTION PROCESS - It is a process plant (or machine) including its control which provides a service to the environment (production of some products).

THROUGHPUT: - The quantity of data a system is capable of handling without delays.

SAFETY - It is reliability in respect to critical failure modes.

STATE INFORMATION - It is concerned with attribute values of objects which are valid for a given time interval.

SYSTEM - It is a collection of entities, organized according to a plan, in order to provide a service to the environment.

RELIABILITY

$$\text{AVAILABILITY} = \frac{\text{MTTF}}{\text{MTTR+MTTF}}$$

λ = constant failure rate

MAINTAINABILITY, M(T) - It is the probability that a system that has failed will be restored to an operational level within the specified time, when the maintenance action is performed in accordance with prescribed procedures.

MEAN TIME TO FAIL, MTTF - $1/\lambda$

MEAN TIME TO REPAIR, MTTR - $1/\mu$

MINIMUM PERFORMANCE CRITERION, MPC - Any level of service above the MPC is success and below the MPC is failure. For reliability equations normally the MPC is of importance.

RELIABILITY, R(T) - It is the probability that a system will function within the specified limits for at least a specified period of time under specified environmental conditions.

RELIABILITY OF COMPONENTS:

COMPONENT	ORDER OF MAGNITUDE OF λ IN FAILURES PER 10^9 HOURS
Transistor	10.....100
VLSI Component	100.....1000
Thyristor	100.....1000
Reed relay	100.....1000
Servo motor	300.....3000
Solder connection	2.....20
Switch (per contact)	30.....300
Plug connection (per contact)	30.....300
One board computer (25 chips)	4000.....40000
Fixed head disc	20000.....200000
Fan	10000.....100000

μ = constant repair rate.

CHAPTER 1

INTRODUCTION

1.1 Background

The new field of distributed computer control systems(DCCS) is currently receiving a great deal of attention and appears to offer many benefits.

However, at present, there are very few, if any, truly distributed process control systems in operation. In order to assess how readily these promised benefits of DCCS can be realized, this project report presents a case study in which the design of a control system is approached from a distributed point of view. The process plant considered is the water treatment plant (WTP) section of a coal-fired power station.

1.2 Overview of the Plant to be Controlled

This section explains the importance of water purity and therefore the importance of the water treatment plant in overall power station.

A thermal power station is basically an energy conversion machine which involves the following simple steps: (see fig.1.1).

a) Chemical energy (coal or oil or nuclear) is converted to heat energy (in the boiler).

b) Heat energy is converted to mechanical energy (in the turbine).

c) Mechanical energy is converted to electrical energy (in the generator).

The coal or oil is combined with oxygen in the air under controlled conditions to generate heat (fire).

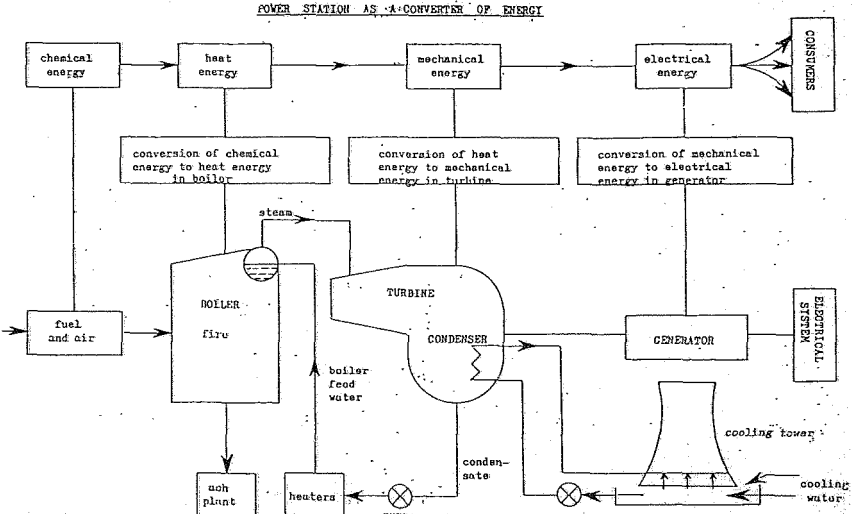

POWER STATION AS A CONVERTER OF ENERGY
chemical energy
heat energy
mechanical energy
electrical energy
CONSUMERS
conversion of chemical energy to heat energy in boiler
conversion of heat energy to mechanical energy in turbine
conversion of mechanical energy to electrical energy in generator
steam
BOILER
fire
fuel and air
TURBINE
CONDENSER
GENERATOR
ELECTRICAL SYSTEM
boiler feed water
cooling tower
condensate
cooling water
ash plant
heaters
pumps
pumps

FIGURE I.1

Water is used as a medium of heat transfer in a thermal power station. The heat generated in the boiler converts water to steam at a high temperature (TH) and pressure (thermal energy). As this steam moves through the turbine, it is converted to steam at a lower temperature and pressure (thermal energy is converted to mechanical energy). Steam is ultimately condensed to water at a low temperature (TC) and pressure. The water is cycled through the system again.

There are three important water circuits in the system. These are listed below:

a) Boiler feed-water

b) Circulating cooling water (CCW)

c) Potable water, fire water and a water circuit for general purposes.

The first water-circuit is the most important for the generation of electricity in a thermal power station. Water circuit (b) is not essential, as dry cooling (or fans) can be used to cool and condense the turbine exhaust steam to boiler-feed water. The last one is used mainly for auxiliary services in the power station. This dissertation will concentrate mainly on (a); (b) and (c) will also be discussed for completeness sake.

The water flows in pipes throughout the whole system. The water contains various chemicals. Chemical deposition inside the pipes is not desirable because it has the following detrimental effects:

a) Reducing the bore of the pipe:

Once the bore of the pipe is reduced, it offers resistance to the flow of water or steam (Fig. 1.2)

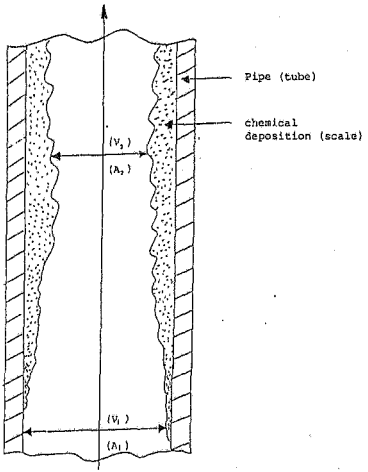


Fig. 1.2

Restriction of water/steam flow due to chemical deposition inside the pipe (Tube).

For a *constant pressure* in the pipe, the continuity equation ($A_1V_1 = A_2V_2$ for P = constant) yields:

$$V_2 = \frac{A_1 V_1}{A_2} \quad m/s$$

This implies that the velocity of the fluid in the restricted area has to be higher than in the non-restricted area if flow continuity is to be maintained.

b) Reducing the *thermal efficiency* of the system:

This phenomenon is shown on fig. 1.3.

The basic equation is: $Q = \frac{T_1 - T_2}{R}$

where Q = *the quantity of* heat transferred from the fire to the water.

T_1 = the fire (or higher) temperature.

T_2 = the *water/steam* (lower) temperature.

R = resistance to heat transfer (due to the scale).

The basic principles *involved are:*

i) the fire radiates heat to the boiler pipe metal.

ii) the boiler pipe metal conducts heat to the *water/steam*

iii) the circulating water takes the heat away.

The basic control variables are:

i) the difference between T_1 and T_2 is maintained by the circulating water/steam If circulation stops, T_2 becomes equal to T_1 and the tube eventually melts.

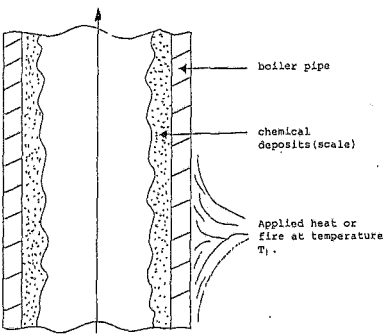


Fig. 1.3

Restriction of heat flow due to scale formation inside a boiler pipe.

ii) Resistance to heat transfer: resistance (scale) prevents the tube metal from transferring heat to water or steam. If the tube metal temperature becomes equal to T_1 the tube melts.

iii) T_2 which is the fixed value: T_1 is controlled by the rate of circulation.

The chemical deposit (scale) is a bad conductor of heat and if this forms on the water side of the tube, due to impure water, the tube will receive the heat, but will not transfer it to the water. The tube will overheat and burst. If circulation of the water is stopped, the tube will overheat and burst.

It is because of the two problems outlined above that the boiler feedwater has to be chemically treated to de-ionize it to a specific purity before it is fed to the boiler. The water treatment plant of a thermal power station therefore forms an intergral and important part of the power station. Control systems for water treatment plants have often been designed before using conventional (i.e. centralized control) methods, and such a plant therefore provides a suitable test case for evaluating a distributed control system design.

1.3 Problems Occuring in Process Control System Design

The main reasons for the decentralization of process control computer systems are the necessity to control plants that are spread over larger physical areas, the growing demand for higher availability and the growing functional complexity of processes. These points influence the structure of a decentralized computer-system for a given process in different ways.

Distributed computer control systems offer many benefits, but coupled with these benefits there are many problems which have to be overcome if these systems are to be used. Some of the problem areas to be tackled are:

1. Communication between the various functional units.

2. The type of architecture that can be supported by this system.

3. The type of components that should be used.

4. Fault tolerance and recovery in the system

In this project report I have chosen to concentrate on four major problem areas that are encountered in process control system design. These four areas are:

1. Remote signal transmission

2. System back-up

3. System information coordination

4. System safety.

1.3.1 *Remote signal transmission*

A communication system forms the backbone of any industrial process control system. In traditional centralized control systems, both the information display/operator interface and control equipments were located in a central control room and the signals to/from individual field-located instruments and actuators had to be gathered at this point. In more modern systems it is possible to physically distribute to some degree the control functions and locate these close to the point of control.

It will however always be necessary to provide centralized information display/operator interface facilities and to inter-connect the various distributed control functions. It is therefore important to examine the problems and trends in remote signal transmission.

1.3.1.1 Cabling considerations

The cost of running cables from the plant area to the central control/equipment room is very high, especially in large spread out continuous processing plants. Any concept that could reduce costs should receive serious consideration.

The most expensive kind of cabling system for a centralized control system is one where, for each signal required from the field or central control area, a cable is run solely to transmit that signal (see fig. 1.4). This is the cabling system presently used in the plant which I describe as a case study in chapter 4.

Remote signal multiplexing (see fig. 1.5) is a system which is far better than the one described above. Instead of running one cable for each signal; two or more signals or messages can be sent over a single channel or set of wires simultaneously. The first published reports (25), (26), (14), (16) and (27) all hinted at the cabling costs savings inherent in the application of a multiplexer concept.

Simon(24) concluded that "there have been dramatic savings in wire and labour costs in the application of field multiplexing systems". These conclusions were drawn in the days before the availability of modern LSI techniques. These now permit computing power to be placed nearer to the process to be controlled. This is what is envisaged for the control system of the water treatment section of a thermal power station.

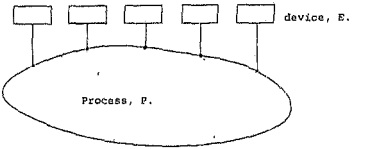


Fig. 1.4
Totally parallel control with parallel individual device,E. One device E is present for every control function in the process,P. Examples of E are: single loop controller, individual manual control device, recorder.

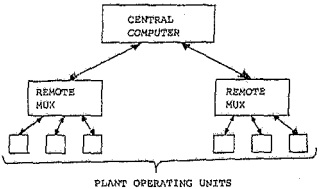


Fig. 1.5
Remote signal multiplexing.

1.3.1.2 Estimates of cable costs savings:

More up to date estimates of cost savings are available where the new technology has cut down on distances (6):

- DEC has built a case around a centralized computer control system of 450 points running on the average 1500 ft. They estimated cabling costs of approximately $390 000. Reducing the distance to 200 ft would reduce the cabling costs to $79 000.

- Hewlett-Packard took note that while cabling costs are rising about 20% per year, computer costs are going down at a rate of 20% p.a. They say that the cabling costs can be as high as two or three times the computer costs.

- "The use of remote I/O (remote multiplexing) eliminates very costly cable runs from transducers to the computer I/O terminals. On a plant such as a hot strip mill, the savings in cable cost can almost completely pay for the computer system" (5).

- In a large batch plant, "Centralization of the computer and an extensive centralized analog and digital back-up (700 analog inputs, 300 analog outputs, 1000 digital inputs and 2000 digital outputs) involves a tremendous amount of cabling (value of single pairs and multi-core amounts to approximately $4 000 000). The time necessary to install all these cables and cable trays will require approximately 50 000 man hours. Using remote multiplexing, it is anticipated that this technique will reduce the total automation investment by approximately 25%" (3). Using DCCS will reduce costs even further.

1.3.1.3 Cable cost estimates for case study:

For the case study presented in chapter 4 there are 350 analog inputs, 200 analog outputs, 5700 digital inputs and 3000 digital outputs. A lot of cabling is involved in this plant. An estimate of the cable costs for all the different types of cables is R2 000.000,00 (=33% of total C & I costs for the plant). This excludes the cost of designing cable trays and pulling the cables.
The amount saved due to the introduction of DCCS cannot be estimated as the components to be used will only become fully available in the next couple of years. The required characteristics of these components are described in chapter 3.

1.3.2 Back-up:

I would like to point out that the type of back-up philosophy decided upon strongly influences the configuration of a give control system.

1.3.2.1 How much and what kind of back-up?

The back-up concept has posed great problems to management, production, engineering and operations. What should be done when the computer fails in a centralized system? Aspects to be considered in this connection are:

a) Safety
b) Equipment utilization
c) Process equipment function (type and degree of intercoupling)
d) Additional control devices
e) Fail-safe modes
f) Operating personnel
g) Economics of materials involved (dumping, re-running, spoiling, etc).

Back-up can be incorporated into the system by providing complete redundancy in control equipment; i.e. conventional analog controllers which the operator(s) can switch on when the computer system is faulty. This makes the computer system design expensive and difficult to implement.

1.3.2.2 Amount of redundancy

Microprocessors/minicomputers are cheaper than and have uptimes that exceed those of the equipment (pumps, valves, vessels, etc.) with which they work. However, when finally the computer does fail, there may be no operating personnel who can easily and quickly effect the changeover to the back-up controls. They will have since forgotten about them because they will have been completely dependent upon computerized control.

Redundancy (active redundancy) is easy and cheap to implement in DCCS using microprocessors/minicomputers. The operator does not have to do anything if one control system fails because the system may be designed with two such systems working in parallel. The system may also be designed such that it works in a degraded mode when such a failure occurs.

It is for these reasons that a DCCS is preferred to a centralized control system.

1.3.3 Information coordination:

A distributed system offers the advantage of being able to tailor the coordination of information on the basis of need, rather than on a take-it-as-it-is basis which is the case for a single central computer system. Having separate processors allows the system designer to provide minimum of communication if the tasks for the processors are carefully thought out.

The exchange and coordination of information can be based upon flexibility provided by the distributed network concept.

The advantages provided by a distributed network concept are:
a) Flexibility in modifying data requirements
b) Reliability in checking software
c) Increased flexibility in control system concept, etc.
These are discussed in chapter 2. A centralized control system does not offer the advantages outlined above.

1.3.4 Safety

Inherent in all of the aspects of distributed systems discussed in chapter 2 below, their advantages, the flexibility, increased reliability, etc. is the all-important aspect of increased safety.

Putting an operating interface out where the action is, allows more control possibility to be exercised there. As Arendt(1) states "... the ability to sustain the process temporarily can be extended to include shut-down procedures or a safe reduced operating leve mode to allow for communication or central controller failure."

Safety can be enhanced if the entire monitoring, display and/or reporting of undesirable trends can be decentralized. Closer observation is made possible in a distributed scheme because a few people are then involved only with what is happening in their particular areas of responsibilities. This permits taking corrective action on the spot.

Berret, Hurd and Williamson also state (4), system intergration "must not proceed so far that safety and integrity are threatened... we need to prevent fault propagation and insure that malfunction or damage to one part of the system will not crash the entire network".

Distributed computer control systems also offer other benefits compared to centralized or conventional computer control systems. I have outlined the main benefits above under the topics: remote signal transmission, system back-up, system information coordination and safety. It is for these benefits that I propose and strongly recommend that distributed computer control systems be implemented in the control system of the case study.

The application of this control concept is discussed in chapter 4.

1.3.5 Fault tolerance and recovery

These are very important aspects of any process control system and will be fully treated in chapter 2.

1.4 Problem Statement

Automatic control of the water treatment plant of a thermal power station, whether achieved by a distributed 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 complex, usually nonlinear, relationships for the translation of the plant 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 processes of the WTP of a thermal power station, 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.

1.5 Objectives

The objective of this study is to explore the issues in designing a distributed computer control system (DCCS) for the water treatment plant (WTP) of a thermal (coal-fired or oil-fired) power station.

There is an existing physical plant which can be divided roughly into eight major systems, each of which comprises of from 4 to 12 subsystems. A detailed discussion of this is given in chapter 4.

From the point of view of control system design, the WTP of a thermal power station is large (about 6 km circumference) and complex (about 10 000 monitor and control variables). The goals of this system are data acquisition and control.

Control system design goals assumed in the present study:

1. All the WTP systems will be operated by a network of microprocessors distributed throughout the plant.

2. In the case of the failure of all or part of the control microprocessor network, the system will continue to operate in a (possibly) degraded mode. In addition, such a loss will not, of itself, cause operations to be aborted. This principle or goal has important implications for the control system design.

3. The hardware and software interface of devices to the WTP control system will be implemented in standard ways. That is, devices connected to the control system must satisfy appropriate connection standards for both hardware and software. The importance of protocol and interface standards is well known in the distributed data processing field.

4. The generation of software and the testing of new devices should be routinely done by the control system users. Hence, the user must have at his hand *comprehensive* debugging and monitoring facilities.

5. The control system *should be capable of modelling* the state of the WTP both in a predictive and a historical sense. This feature is more than useful, it is necessary. The WTP is complex and fragile. An operator needs to know the consequences of actions *before the actions are implemented*. Such proposed actions can be tested in the WTP model, and thus the operator can know about the *implications*. Facilities are also required to analyse and identify the causes of certain events, e.g. why a particular alarm comes up. This is known as the historic ' WTP model.

CHAPTER 2

SURVEY OF DCCS TECHNOLOGY

2.1 Working Definition of a DCCS:

It is difficult to formulate a good definition for a DCCS. However, Enslow (21) proposed a working defintion that had five components, all of which he considers necessary to define a distributed system.

These five components are:

i) "A multiplicity of general-purpose resource components, including both physical and logical resources, that can be assigned to specific tasks on a dynamic basis. Homogeneity of physical resources is not essential.

ii) "A physical distribution of these physical and logical components of the system interacting through a communications network. (A network uses a two-party cooperative protocol to control the transfer of information.)

iii) "A high-level operating system that unifies and integrates the control of the distributed components. Individual processors each have their own local operating system, and these may be unique.

iv) "System transparency, permitting services to be requested by name only. The server does not have to be identified.

v) "Cooperative autonomy, characterizing the operation and interaction of both physical and logical resources."

I propose that we accept Enlow's working definition as the base point for this discussion.

Enslow (21) characterizes distributed data processing systems in a three-dimensional space that illustrates the concept of a multidimensional parametric space. Enslow's figure, reproduced here as fig.2.1 characterizes a distributed data processing sytem in terms of the decentralization of hardware, control and data base.

2.2 Factors to be Considered in the Selection of a System Architecture

2.2.1 Advantages of distributed systems

Although many advantages are claimed for DCCS there are also many obstacles. The use of DCCS assumes that somehow distribution of functions into small, separated units will be simpler, then combining these functions in a single unit. It is essential to recognize that the coordination of distributed elements will in fact be more difficult because information is much less accessible from unit to unit. There are four basic reasons and a series of related lesser reason why there is such an interest in distributed systems.

These can be enumerated as:

1. Ideally, distributed systems can be more reliable than a centralized or "single block" system. The effects of failures are localized and easily corrected. They allow graceful degradation of the process control system, i.e. failure in one or more components or subsystems will not cause the entire system to fail. These systems are as reliable and correctable as analog control when the basic level of modularity is correctly chose.

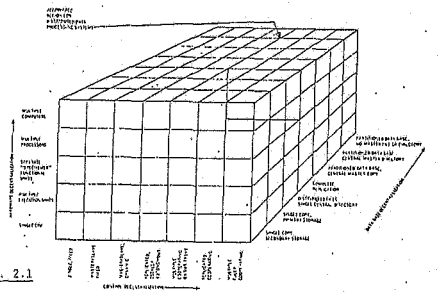


Fig. 2.1

Dimensions characterizing distribution.

2. A general network, as compared to a hierachy or a similar structure allows redundant (reliable) communication.

3. Ideally, distributed systems allow the improvement or evolution out of which the application is built.

4. When compared to centralized organizations, a well-designed network or hierarchy can allow a reduced system wide data communication rate by localizing function and data requirements. This also contributes to the localization of the effects of communication failure. This is of interest because most existing process control distributed system proposals are highly centralized (about a highway).

5. Distributed systems allow the system hardware to be visibly structured, for operation and maintenance, on functional grounds. Failures can be recognized in terms of errors of function rather than abstract errors of computation.

6. Moreover, the functionally based design approach minimizes the necessity to account for the effects of confusing interactions between the parts of the system.

7. A general network attitude allows (over other more fixed organizations) flexibility and evolution of the communication relationships. For instance, the network allows the physical movement of the command centre. This is much more difficult in a fixed hierarchy.

8. The structure of hierarchical and distributed systems corresponds to traditional analog control structure and human organizations familiar in process control; operating personnel are used to them.

9. Existing separated systems may, by later decision, come to be coordinated by connecting them with communication links (if the elements were designed to allow for this possibility). Suddenly, a distributed system has evolved.

10. Ultimately, distributed systems should simplify installation and cabling, reducing the related costs.

These systems lead to low cost due to:

a) Simplified hardware configuration packaging since processors need not be large due to the reduced processing requirements of each partial process, as mentioned in 5. , above.

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, as mentioned in 5. , above.

c) Large scale integration.

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 or components may be added to the system without drastically interfering with the functions of the rest of the system. (point 7 above).

f) It is cheap to maintain a reliable system (see the trends indicated in fig.2.2 below).

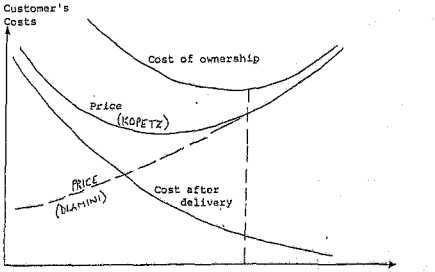


FIG. 2.2

Reliability and customers' costs.

11. Structuring and parametising should be easy to learn, *especially without programming knowledge of process computers.*

12. Simple user-orientated solution even of unusual control tasks, e.g. by higher algorithms.

13. High safety and availability, variable according *to requirements.*

14. Compatibility for the trouble-free incorporation of hardware innovations, e.g. new microprocessors, fibre-optic conductors.

15. Distributed systems are very effective process control systems, and their effectiveness depends on a variety of factors shown on Fig.2.3 below.

There are many more reasons why distributed systems are a viable proposition, but I will not list them here. The list given above *suffices for the present discussion.* To sum up, the present-day technical reasons for distributed systems are reliability, maintainability, flexibility and cost. These four factors are all intimately related. Ideally, the greatest flexibility offers the greatest variety of recovery responses to the greatest variety of element failures.
Badly designed *flexibility leads to failures* or unreliable application design. An unreliable or non-easily maintainable system may be very costly in a variety of ways, some of which may be -

- loss of production due to failures
- loss or destruction of equipment/life due to system failures
- long *maintenance times*
- and many others.

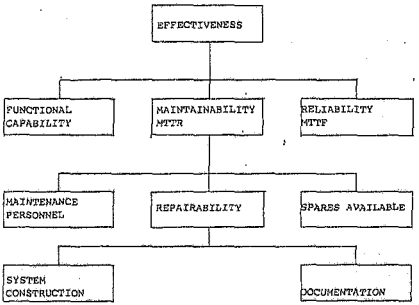


Fig. 2.3

Interrelationship of functional capability, maintainability and reliability.

2.2.2 Reliability and Maintainability

The success of every organization, which operates an intergrated information system, is critically dependent on the reliability and maintainability of the distributed system.

2.2.2.1 Reliability

The reliability requirements on the different functions or components or units of a distributed system will very. In order to achieve a solution which is optimal from the point of view of cost effectiveness, it must be possible to design different subsystems to different levels of reliability, i.e. reliability has to be a design parameter. This is achieved by incorporating redundancy differently to different components.

2.2.2.2 Maintainability

The maintainability of an integrated DCCS has two aspects: hardware maintenance, i.e. repair of the hardware; and software maintenance, i.e. modification of the software. The hardware maintainability can be considered a by-product of the reconfiguration capability needed for fault tolerance. The dynamic replacement of faulty units requires an automatic error isolation to the smallest replaceable unit (SRU) which is the most time consuming step during manual maintenance. The software maintainability, i.e. error correction, modification and functional enhancement of the software, is a permanent requirement over the lifetime of the system. It has been shown (11) that much more effort is needed for the continuous maintenance of the software over the lifetime of a successful system than for its initial development. In order to make a system maintainable from the point of view of software a decomposition in self-contained autonomous subsystems with clearly defined interfaces is required (16).

It must be possible to test each of these subsystems independently and monitor the information to and from each subsystem without undue technical overhead.

2.2.2.3 Possibilities of increasing the reliability of operating systems

The reliability of an operating system can be enhanced/ increased by incorporating one or two or all of the following, in a distributed control system.

i) Additional individual manual control devices.

ii) Additional local operating system (decentralized auxiliary control board).

iii) Redundancy.

2.2.3 Sources of system unreliability

The obvious sources of unreliability in a system made of completely debugged elements are poor application design and element failure. More generally, the basic barrier to reliability and to flexibility is system complexity, particularly forms of complexity which hide the relation between the control system and its operation and purpose. Several specific barriers to reliability are worth mentioning:

1. Multiple loosely correlated control functions unnecessarily combined in a large computational unit or box. In this case, the failure of the unit or its communications or central control loses function unnecessarily.

2. Separation of tightly correlated functions unnecessarily in separate units. Here the loss of one function in one unit is likely to be as serious as the loss of both functions (i.e. both units), a weak link phenomenon.

3. System complexity which encourages application errors, and makes failure diagnosis and maintenance difficult.

4. Unstructured system and system component design.

5. Insufficient back-up.

6. Back-up complexity. Back-up depends on the assumption that the switchover mechanism (where standby redundancy is used ; active redundancy is generally better) is much more reliable than the control element.

   When this is not the case, back-up hurts more than helps.

7. Back-up by elements/components which operators only encounter during crises and are likely to misuse through unfamiliarity.

To sum up, the principle sources of unreliability in a system are complexity, mismatch between design purpose and appearance and inconsistencies in the operational characteristic of system components.

2.2.4 Disadvantages of distributed systems

2.2.4.1 High-level operating system:

The high-level operating system is a key ingredient in the distributed system. Its design must take into account several characteristics and problems.

1. The classical design for operating systems, as it has developed, assumes the availability of a large amount of system information. Although the completeness and validity of information about the work being presented by the user is questionable, the operating system is usually assumed to have access to complete information about the environment in which it is functioning.

This is not the case in a distributed system: complete *information about the system* will never be available. The resources provide a service, but they may, either intentionally or unintentionally, shield information from outside inspection.

2. In distributed systems, there will *always be a* <u>time delay</u> in the collection of information about the status of the system components/units. The ramifications of these time delays are extremely important. In a conventional centralized processor, the operating system can request status information, being assured that the interrogated component will not change state while *awaiting a decision based on that status* information, since only the single operating system asking the question may give commands.

   In a distributed system, the time lags that occur can become significant; as a result, inaccurate (badly out-of-date) information can be transmitted because the autonomous *component proceeds along* its own path.

   For distributed systems, it will be essential to raise the degree of paranoia (abnormal tendency to suspect and mistrust-information/data in this case) of the system designer to a much higher level than for *centralized systems. The system must* be designed to work even with erroneous or inaccurate status information.

3. A further complication with regard to system information available is the possibility of variations in *the information* presented to different system controllers. These *variations may be a* result both of time delays and of different controllers. As Le Lann (17) has observed "This absence of uniqueness, both in time and in space, has very important consequences".

This is not the case in a distributed system: complete information about the system will never be available. The resources provide a service, but they may, either intentionally or unintentionally, shield information from outside inspection.

2. In distributed systems, there will always be a time delay in the collection of information about the status of the system components/units. The ramifications of these time delays are extremely important. In a conventional centralized processor, the operating system can request status information, being assured that the interrogated component will not change state while awaiting a decision based on that status information, since only the single operating system asking the question may give commands.

   In a distributed system, the time lags that occur can become significant; as a result, inaccurate (badly out-of-date) information can be transmitted because the autonomous component proceeds along its own path.

   For distributed systems, it will be essential to raise the degree of paranoia (abnormal tendency to suspect and mistrust-information/data in this case) of the system designer to a much higher level than for centralized systems. The system must be designed to work even with erroneous or inaccurate status information.

3. A further complication with regard to system information available is the possibility of variations in the information presented to different system controllers. These variations may be a result both of time delays and of different controllers. As Le Lann (17) has observed "This absence of uniqueness, both in time and in space, has very important consequences".

2.2.4.2 General control problems

High level operating systems as described here are highly nonhierarchical - that is, they are single-level and have no internal master/slave relationships. This characteristic, combined with component autonomy, greatly aggravates the control problems.

1. Even if autonomous multiple components are cooperating, the propability of simultaneous conflicting actions is much higher than in hierarchical systems.

2. Also, synchronizing the actions of the various controllers in the system is much more difficult, because of the presence of appreciable time-lags.

3. The problem of deadlocks or infinite cycles within the system is quite different from that associated with other systems. Some proposals call for a supervisor (an outside third party) to solve this problem: however, such a supervisor would have to be transient, since the presence of a *permanent supervisor would* denote an unacceptable degree of hierarchical control.

4. *Any system exhibiting monolithic, autonomous* control presents completely new problems in system scheduling. A request for service in a nonhierarchical system might well result in an initial denial of that service by all physical resources. In that instance, the requesting entity might initiate an evaluation of relative priorities between the new request and currently executing tasks, followed perhaps by bidding (priority adjustments) and decision-making. The efficient execution of this procedure is one of the most important functions of the high-level operating system.

5. Resource management in a distributed system is a multidimensional job. Thus far, very little work has been done on the aspects of resource management that apply specifically to distributed processing systems. However, low-level functions are quite similar to those performed on uniprocessor; they include physical resource allocation and, management of those facilities required by a process after it has been scheduled on a particular system component.

   Before that can be done, however, the required resources may have to be assembled at one location, or linkage mechanisms established so that they can be used remotely. The problems that have to be addressed in that process are locating the resources, determining which components are suitable, and determining the best way to move the resources to the selected location. At an even higher level is the scheduling problem, determining when a function should be initiated or terminated.

When all of these problems and their possible solutions are compared to similar problems and solutions encountered in uniprocessor systems, the major factor aggravating the distributed system problems is seen as communication within the distributed system, which is asynchronous with respect to the detailed execution of the functions, and which exhibit time-lags in addition to the communication processing time itself. Uniprocessors cope with many of the problems with semaphores, flags, lockout gates, or time-outs. To attempt to do this within a reasonably complex distributed system requires too much time, in the sense that such processes reduce the throughput rate of the system. The transit time for signals transmitting the semaphores is of the order of 100 mS. In addition to the lowering of performance, the reliability and the robustness of most of the uniprocessor solutions are in doubt.

Distributed computer control systems are still used despite these disadvantages, because most of these shortcomings can be resolved to an acceptable level; and that most process control systems are slow and also do not require complex back-up systems (i.e. not critical). In addition to this, DCCS offer more benefits than disadvantages.

2.3 Requirements of DCCS

2.3.1 Functional requirements:

The background of the growing acceptance of the first generation of DCCS in the various process industries has been the solution of drawbacks arising out of the conventional types of central control systems (27). In such a conventional system the operator has to collect the plant data from physically separate positions of a large control panel. Since only a small amount of automated information processing is provided by such control systems many routine tasks have to be performed by the operator.

Already the first generation of DCCS has helped the operator by automating some of these routine tasks,

e.g. - preprocessing of the plant data to support the decision-making of the operator.

- execution of more advanced algorithms to handle deadtime and delaytime.

- automation of some of the safety procedures during start-up and shut-down of the plant.

- automation of data acquisition, recording, tracking and retrieval.

In the next generation of DCCS, the support of the operator at the man-machine interface (MMIF) must be intensified by providing the following additional functions:

- the images of the conventional analog instruments, which are produced on the colour CRT, will have to be replaced by more meaningful patterns to ease the human perception.

- the flexibility of the man-machine interaction has to be improved in order to give the operator more direct access to the information needed at a given point in time.

- some parts of the alarm analysis have to be automated.

- provision of more advanced control algorithms including multivariable control.

- provision of simulation packages of the process in order to help the operator in the evaluation of alternative courses of action; mentioned above.

- provision of software packages which integrate costs information (e.g. material, costs, energy) with the technical information in order to guide the operator in finding the most cost effective operating point of the process.

- provision of software packages which support the plant maintenance.

2.3.2 Communication system requirements:

The downward integration of a DCCS to the sensors and transducers will eliminate the need for some of the conventional analog circuitry for information processing in the process peripherals.

It will also be necessary to include the operational control of the electromechanical devices (e.g. electrical switchgear, pumps, compressors, etc) in the scope of the DCCS. In order to completely replace the conventional wiring between the intelligent instruments and the control room, communication channels with the following characteristics are required:

1. The reliability must be at the least equal to the reliability of the conventional two wire systems.

2. The real time constraints of the problem must be considered, i.e. the delays introduced by the communication system must be short and have a guaranteed maximum value.

3. It must be possible to reconfigure and expand the communication system dynamically and incrementally, i.e. the communication system must have high extensible transport capacity.

4. The information which is transmitted across an interface between two subsystems must be observeable without undue technic[illegible] overhead.

5. The capacity of the communication system must be rather high since a lot of sensors and actuators may request simultaneously message-transmission - it is typical for technical processes that in critical situations important messages originate at many points simultaneously.

6. Additionally, the communication system has to provide logical means to coordinate and synchronise tasks that are being performed by different processors.

This is not so trivial since different processors use different time bases and the transport of synchronization messages requires a considerable amount of time. It can be seen that the communication system fulfils an important coordinating function for the decentralized computer system.

7. The user must communicate with the system by directives containing service names only. My criterion of system transparency makes unnecessary and perhaps impossible the user designation of the system component offering a desired service. However, this requirement introduces new problems of system failure and user error *detection, since no one processor* can establish whether the service requested can be provided an[illegible]re in the system, or even whether it is legal.

The above indicates the central role of the communication system. The most important requirements can be summarized as follows:

- the reliability of the communication system must be very high.

- mechanisms have to be provided in the communication system (and rest of the system as a whole) that allow an uninterrupted monitoring of *all resources (computer* and links) and the scattering about of the resulting status information.

2.3.3 Components/functional units requirements

The components have to be supplied as far as possible as standard *modules, namely: smallest replaceable* units (SRUs). This results in cost effectiveness in the system: mean time to repair (MTTR) is reduced and spares are readily available. Components required in a DCCS are discussed in chapter 3 below.

2.3.4 Additional Requirements

1. Graceful degradation in its ideal form means that a system having n processors would still execute all tasks even if m processors fail (n>m) - with the reduced performance $\frac{100 \times (n-m)}{n}$ %.

   The realization of this requirement is only possible in the *reduced form* in that all tasks from one failing processor are taken over by one or more back-up-systems.

   The processor that needs to take over needs at least the following data:

   - formation about the state of the task.

   - formation about the state of the process.

   - after the failure: access to the process-peripherals of the failed system.

2. Another requirement, an additional expense: the process peripherals have to be connected not only to one but to several processors. This is not for signals from sensors but somewhat involved for the output lines. Perhaps the failed processor doesn't recognize its failure and still tries to control the output lines probably in switching *incorrect information*. There must be a way to safeguard this possibility.

3. Even if no back-up systems philosophy is implemented very *effective* diagnostic tools and methods for error recovery become necessary. Consider a computer system with nodes distributed over an area of some square kilometres - if anywhere a component fails it must be possible to localize it exactly and to define the type of error otherwise the mean time to repair (MTTR) becomes too large.

These methods may be integrated in the message communication system.

4. Another requirement is the access to central resources that cannot be completely dispensed with even in decentralized systems.

   Examples are:

   - central data bases. Mass memories will be (because of environmental conditions) concentrated at one or a few places. This means a lot of traffic for the communication system and some additional coordination tasks.

   - Fast specialized processors like array processors may just be installed once.

   - Operator consoles have to be connected to one or perhaps two processors - all messages about the *state of the process and* its control system have to be routed to these processors.

   Central resources are responsible for additional communication traffic and may require large transport facilities.

5. Finally, another requirement is the need for a standard-interface in order to connect computers of different suppliers to one decentralized system. All distributed systems offered on the market today have only one common feature: They are incompatible with all others.

   The reason is not because the suppliers are interested in preventing (although it might be the case at times) the coupling of different systems but the difficulties in defining a common standard.

The definition not only has to include the physical interface but also the logical one - in the computer network terminology the word "protocol" is used for a set of rules established to create a mutual agreement amongst all parties sharing a communication structure.

2.4 Criteria for Distribution

This basic structural problem unavoidably brings up a further question: according to what criteria should the system be distributed? Should these criteria be process orientated, should they be orientated according to the capabilities of the highly integrated technology of microprocessors, or according to the special characteristics of a serial data transmission system with a data bus, or according to a special monitoring and operating system with video display units.

It goes without saying that these criteria must be predominantly process orientated if the danger of becoming caught up in modern trends of short-lived momentary solutions is to be avoided. It is surprising to find that many modern process control systems are constructed to a considerable extent dependent on the special capabilities of one semiconductor technology or another, and much less - if at all - according to the requirements of the process, or on safe, comprehensible, economic and advanced process operation. It is difficult to assess this and to evaluate it with respect to a given system; it is finally also a question of subjective judgement. In this way, for example, one particular user can be completely satisfied if the functional units of his distributed control system are only capable of closed loop control, if it is the case that interlocking and sequencing are little or not at all used in his process, such as in a refinery.

On the other hand, the manager of a power station is responsible for a complex process with strongly *intermeshed control loops*. If he has functional units only capable of closed-loop control then he must realise the open-loop control either by connection to control units which may be foreign to the system, or by using a superimposed process computer. In the latter case this is resorting to conventional direct digital control (DDC) - exactly that which one desired to omit in the distributed system. Such a manager could reach the very plausible opinion that a functional unit in a distributed control system must be absolutely capable of open and closed loop control so that he can fully utilise the advantages of decentralisation discussed in chapter 2.2.1 above.

2.5 Communication System:

With the continuing decline in the cost of computing, we have witnessed a dramatic increase in the number of independent computing systems used for scientific, computing, business, process control, word processing and personal computing. These machines do not compute *in isolation*, and with their rapid increase comes a need for suitable communication networks - particularly local computer networks (or local area networks -LAN) that can interconnect locally distributed computing systems. The local computer/processor must process sensor information, and either take direct action and control valves, switches and pumps, or display the information and turn on alarms where operator control is indicated.

2.5.1 Data acquisition

Process control is a system application characterized by a large number of analog inputs distributed over a wide area, and a data acquisition system for transfer of information from a variety of sensor types to the computing system (local or central).

The classical system approach to data acquisition in process control involves analog signal flow between sensors and computer (fig. 2.4). A typical sensor, such as a thermocouple, supplies a voltage between 0 and 40mV. Because this low level signal is unsuitable for long distance transmission, each sensor or transducer connects to a signal conditioning circuit that translates the signal, typically to a 4....20mA current loop. Analog output of each signal conditioner is fed via a seperate cable or twisted pair to a central analog to digital converter that supplies digitized signals to a computer and to other devices such as displays, limit alarms, and star configuration, with each sensor or point having an individual connection to the central location.

As a result, wiring cost is high, in many cases accounting for 30 to 50% of total system cost. Cabling costs have been discussed in chapter 1 above.

Recent advances in semiconductor technology have made possible monolithic, accurate, low cost analog to digital converters that overcome fundamental limitations of the classical system. The cost of 8- to 12-bit conversion at low speeds no longer requires that many analog inputs share the converter. Connecting a converter to each transducer (fig. 2.5) results in two major benefits. First, the signal conditioner circuit can be simplified or eliminated because the converter is located close to the transducer. Transducer output can either be converted directly, or be conditioned by a single stage amplifier. Secondly, data transmission becomes a totally digital process, with inherent noise immunity and low cost.

A practical future system needs a data transmission link to provide data integrity at minimum cost. For this link, consider a serial versus a parallel structure.

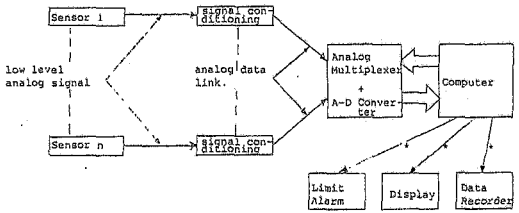


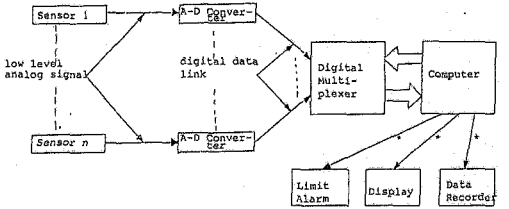
Fig. 2.4

Classical data acquisition system. Low level analog signal from each sensor *must be changed to 4....20mA* (or 0....20mA for some other abbreviation for Companies) analog signal conditioner and transmitted over individual lines to central shared ADC.

The parallel bus is suitable for short distance, high data rate communications. However, process control systems require long distance, typically low speed data links, which indicate implementation of a serial data transmission concept. Whenever data rates permit, serial transmission should be used to reduce cable and wiring costs, allow use of common carrier facilities where appropriate, and facilitate system expansion. After a serial data link is chosen, other design elements such as data rate, 2- or 4 wire, half-or full-duplex, and protocol structure must be decided upon. Obviously, in developing systems for distributed data acquisition and control, it is important that all key elements pertinent to the serial data link be standardized, and that the designer be provided with a wide choice of components and subsystems from various sources, all of which are compatible in data communication.

The serial data link allows the designer to consider a bus or multipoint system configuration, instead of the point to point system. The former is discussed in chapter 2.5.3 below (under LAN). Other implications and possibilities opened up by the bus or multipoint system configuration are a distributed processing concept and a variety of protocols discussed in chapter 2.5.3.1 below.

The practical experience with process control computer projects in industry indicates the growing tendency to consider control in terms of total integrated systems control. Fig. 2.6 below shows the structure of a distributed integrated information processing system. The boxes shown in the diagram are clusters of the different sections of the sytem which have been partitioned according to criteria discussed in chapter 2.4 above. Each box/station/cluster is a computer together with an interface to the communication network.



* Parallel interface.

Fig. 2.5

Alternative data acquisition system. ADC of line driver at each sensor site can eliminate sig.. ' conditioner and transmit information to host computer via digital link.

Each computer (processor) executes a given number of processes. Interactions of processes executing on the same processor is of no interest from the network point of view.

2.5.2 Networks

Kopetz (15 - chapter 3) discusses networks and states the objective of a network as being "to provide access paths for the communication of end users". He defines an access path as a sequence of functions that makes it possible for one end user to communicate with the other *in spite of:*

- various types of transmission errors (collision fragments, etc.)
- differing line speeds
- different data representation (e.g. manchester coding, etc.)
- different name spaces (flat name space or not, etc).

Kopetz (15) defines an end user as:

- a terminal user or
- remote application process or
- data bank process or
- data source (e.g. sensor) or
- data sink (e.g. actuator).

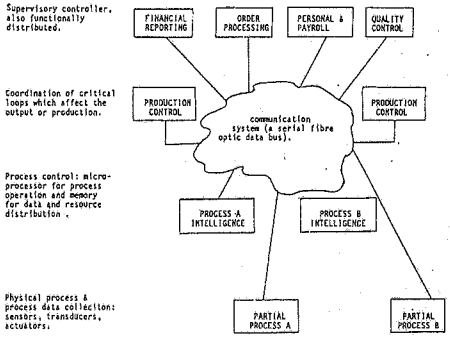


Fig. 2.6

Structure of a distributed integrated information processing system.

In the same book he states the characterization of networks according to criteria such as:
networking technology, topology (bus, ring, etc.), geography (local area network -LAN, long haule network) control, transmission media (cable, satelite links, etc.), ownership (private or public), type of application (process control, banking, etc.).

2.5.2.1 Networking technology

Kopetz (15) enumerates four technologies namely,

- direct physical connection of processors (permanent)

- circuit switching: temporary physical connection

- datagram (or message switching): this way, the information/ data gets to its destination, but there is no acknowledgement on the end user level.

- virtual circuit: virtual connection between end users including flow control and acknowledgement.

Each one of these networking technologies is good in its own right, and their use depends on the application at hand.

2.5.2.2 Networking topology

There are many different topologies which a system can assume. They are: star, hierachy, mesh, bus (with and without a bus controller), ring, etc. For distributed systems a suitable one is the bus topology without a bus master.

2.5.2.3 Network control

Three types of network control are enumerated below:

- centralized: there is one controller/master in charge of the access to the network.

- decentralized deterministic: the access to the network is not random.

- decentralized probabilistic: the access to the network is random, and depends on some probability function. This is what CSMA/CD is based on.

2.5.2.4 Flow and congestion control

Kopetz (15) discusses flow and congestion control.

2.5.2.4.1 Flow control

Control of the flow of information such that the sending discipline does not outpace the receiving disipline. Examples are the par protocol, sliding window, etc.

2.5.2.4.2 Congestion control

Control of the flow of information such that the transport capacity of the network is not exceeded.

Every network with bursty traffic requires flow control and congestion control mechanisms. A congestion control performance trade off graph from Kopetz (15) is reproduced below, Fig 2.7.

2.5.2.4.3 Naming and addressing

"Naming" is a very important aspect in distributed systems. It determines the type of addressing that should be implemented, how the system should be expanded and how it should be maintained.

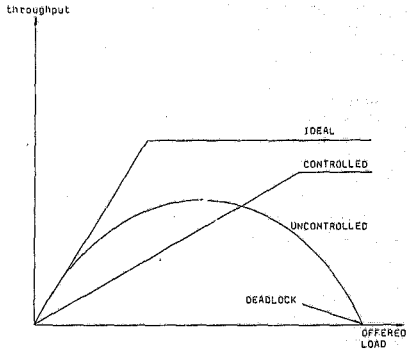


Fig. 2.7

Congestion control performance tradeoff's

What is it, therefore, that should carry a name?
Is it

a) the application process: - this implies that the application process *can be connected anywhere* in the network and the other processes can communicate *with it*.

b) the access point in the network (port)
   i) physical name: - this specifies the physical location of the connection point of the process in the network. This is very advantageous for maintenance.

c) the physical location of an object in the network:- this specifies the physical location of the process in the network, e.g. a tapping point on the 6½m level of a boiler, etc.

d) the access path to an object:-the name *is given to* the access path to the process.

Points a), b) and d) are illustrated on fig. 2.8 and point c) *on fig. 2.9 below.*

The name space design determines
a) the visibility
b) the accessibility
c) the interchangeability of objects

This is illustrated on fig. 2.10 below.

If thermocouples T_1, T_2, T_3, T_4 are not named, but only T is named that is the name space is closed for T_1, T_2, T_3, T_4 then T_1, T_2, T_3, T_4 ... changed without affecting the whole process.

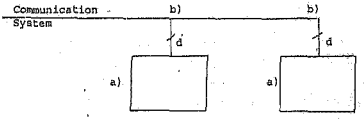


Fig. 2.8

Illustration of points a),b),d).

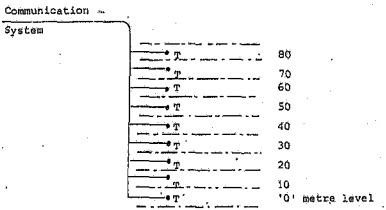


Fig. 2.9

Illustration of point c) - temperature T at 30m level, of a vertical boiler, for example

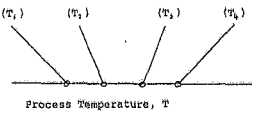


Fig. 2.10

Example: Temperature sensor:

Generally, the local content (T_1, T_2, T_3, T_4) should be hidden, i.e. we are not interested in T_1, T_2, T_3, T_4 - we want to know the process temperature T (Flat name space). The maintenance man would like to know T_1, T_2, T_3, T_4 and T_1, T_2, T_3, T_4 are used to calculate T.

A "clean" name space would be one where T_1, T_2, T_3, T_4 and T are named to cater for both the maintenance man and the "process control" man.

The scope of name spaces can be:

a) global: - used for process control calculations.
   or
b) local: - used for access or maintenance.

The structure of name spaces implies assumption ut the system structure: -

a) flat name space; - hidden details, impl' (..., physically distributed.
b) hierarchical name space: - not fully physically distributed system.

2.5.3 Local area networks - LAN

Networks linking machines together have been around in one form or another for a long time. Data communications began with Samuel F.B. Morse's invention of telegraphy some 140 years ago and expanded with the invention of the teletypewriter around the turn of the century. Computers have been communicating with each other and with remote terminals for a quarter of a century.

Earlier systems communicated over long distances; the communications were fixed between two points, and once the link was set up, it remained established for a relatively long time.

In modern process control systems the entities are confined within a range of a mile or two. The different entities want to access each other all the time: a long distance communication network and a permanently established link between any two entities, are therefore, not required.

A local area network can be defined as a communication network which covers a distance shorter than that covered by a long haule network but longer than that covered by a computer bus, etc. Kopetz (15-p5/1-5/2) compares a local area network with a long haule and a computer bus network. While there is no single definition for a local area network, there is a broad set of requirements:

a) relatively high data rates (typically 1 to 10M bits per second).

b) geographical distance spanning about one kilometre (typically within a building or a small set of buildings);

c) ability to support several hundred independent devices;

d) simplicity, or the ability "to provide the simplest possible mechanisms that have the required functionality and performance";

e) good error characteristics, good reliability, and minimal dependence upon any centralized components or control;

f) efficient use of shared resources, particularly the communication network itself;

g) stability under high load;

h) fair access to the system by all devices;

i) easy installation of a small system, with graceful growth as the system evolves;

j) ease of reconfiguration and maintenance; and

k) low cost;

l) accessibility, to facilitate interconnection of entities to one another with simple addressing;

m) standardization of signal exchange methods, i.e. communication protocols.

2.5.3.1 Types of LANs and the methods of accessing the LAN

Kopetz (15-p.5/4 to p5) discusses different types of local area networks such as: the ring system and single channel (bus) system. He also discusses the characteristics of the ring system and the type of control (token-connector, contention and fixed slots-TDM or FDM) of the ring system.

Kopetz (15) discusses the problems to be solved in a bus system and the types of access to a bus system communication medium namely: fixed assignment (FDM or TDM), demand assignment (central or distributed) and random (random access, slotted aloha, pure aloha, persistent CSMA, slotted non-persistent CSMA) access or a combination of the above. He also discusses the problem associated with each communication bus access technique.

The carrier sense multiple access with carrier detection (CSMA/CD) and the ethernet system (as CSMA/CD is applied to it) are discussed in 15. Kopetz (15) compares the efficiency (throughput) of the different random access systems under different load conditions-p.5/16; and the delay thereof under different throughput conditions.

At the end of chapter 5, Kopetz (15) mentions the IEC bus (IEEE 488) for the connection of instrumentation to a computer/processor.

The throughput of a network can be described as referring to how quickly data can be transmitted among terminals and peripherals. Network throughput can be expressed by a load/response time curve such as shown in fig. 2.11 below. Most networks are efficient for as long as the load is light to moderate. Once the knee of the curve is reached, service deteriorates rapidly, and transmission delays result.

2.5.3.2 Bus system of a LAN:

The bus system of a LAN is shown in fig. 2.12 below.

2.5.3.3 Transmission media:

Another important consideration in implementing LAN is what type of transmission medium will be used to interconnect the processes. Currently, coaxial cable and paired wire have the inside track, with a relative newcomer, fibre optic cable, rapidly gaining acceptance.

Paired copper wire has the advantage of being low cost, and if the distance between processes is short, it is capable of reasonably fast transmission speeds - about 1,5 M bits/sec over a range of about 1.5 km. Coaxial cable is capable of much greater transmission speed and currently considered to be the most desirable medium. Fibre optic has the greatest data- carrying capacity of all, but it has other limitations, the principal ones being that it is expensive and more costly to connect to multiple processes.

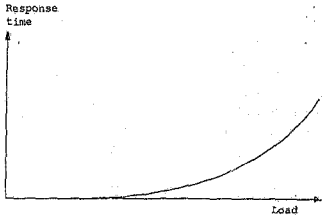


Fig. 2.11

LOAD/RESPONSE TIME OF A TYPICAL NETWORK

Local area network terminology are listed in the glossary..

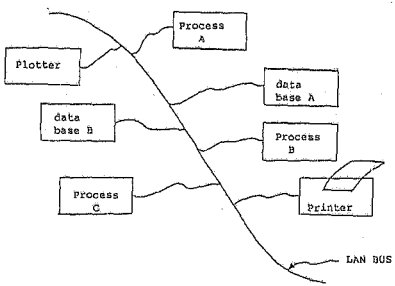


Fig. 2.12

SCHEMATIC OF THE LOCAL AREA NETWORK

2.5.3.4 Interprocess communication:

Kopetz (15) defines interprocess communication as dealing with the exchange of information between sequential processes executing in parallel, where the information exchanged in general comprises

a) synchronization information.

b) a data structure.

He defines a message as an indivisible (atomic) data *structure, which is formed* for the purpose of interprocess communication. He defines two events which are *associated* with the transfer of a message namely:

a) the event of sending the message, Es

b) the event of receiving the message, Er

Kopetz (15) refers to a message with no data structure as a signal.

He also discusses the classification of the interprocess communication mechanism according to:

a) who is the partner?

b) communication pattern: 1-1,1-n,n-n (3 patterns).

c) assumptions about the reliability of the medium.

d) conditions for the acceptance of a message by the receiver.

e) the meaning of the acknowledgement.

f) syntactic structure.

There is therefore a total of 3^6 interprocess communication primitives.

2.6 A Distributed Computer Control System (DCCS) Architecture

2.6.1 Selection of architectures for DCCS in real time applications:

To achieve an acceptable balance between cost and performance in designing distributed systems, architectures must be selected to match the specific requirements of each application. Because of the long lead time in large projects like nuclear power plants, a decision to exploit distributed systems cannot wait until all pertinent information becomes available. Using established concepts, a number of "building blocks" or clusters are defined as necessary components of [illegible] representation from which system characteristic [illegible] derived. A generalized design sequence which [illegible] identify three major problem areas : process [illegible] clusters, data communications and data base. The design of a distributed data base is the area of most concern as it is the least understood and is complicated by its interaction with the selection and implementation of a partitioning or "clustering" strategy.

2.6.2 Characterization of architectures:

A system should ideally be characterized completely using a minimum number of "independent components" or "building blocks". The set of selected building blocks together with the method of interconnection and the resulting interaction define the system architecture. While in practice completeness and independence are rarely achieved, the system architecture can be tailored to match the specific requirements of each application and to strike an acceptable balance between cost and performance.

2.6.3 A Generalized design sequence

Based on the knowledge of the design processes for centralized systems and data communication networks, a generalized design sequence has been derived for distributed systems. (see fig. 2.13), to illustrate the applicability of established methods and to identify new problem areas. The first step in the sequence, the definition of system requirements based on some understanding of the application, is common to any design. However, the selection of a partitioning strategy, an a priori decision in centralized systems, becomes a critical step in distributed architecture design. The choice depends upon a thorough understanding of the application as it requires the decomposition of the global problem (process control), into its elements (local data acquisition, processing, storage, manmachine interactions, etc.) It has a direct influence on the adequacy of the design and largely dictates the relative difficulties experienced in the three major design areas:

a) processing clusters

b) data communications and

c) data base.

Eventually, the three areas combine to form the actual system. Overall evaluation is performed to assess the strengths and weaknesses of the entire design. Innumerable interactive loops occur in the sequence but only the major ones are shown for the sake of clarity.

Appendix 2 gives a more detailed discussion of the subject "A distributed computer control system (DCCS) architecture". This gives more information on system building blocks, processing clusters, data communication, data bases and control hierachies.

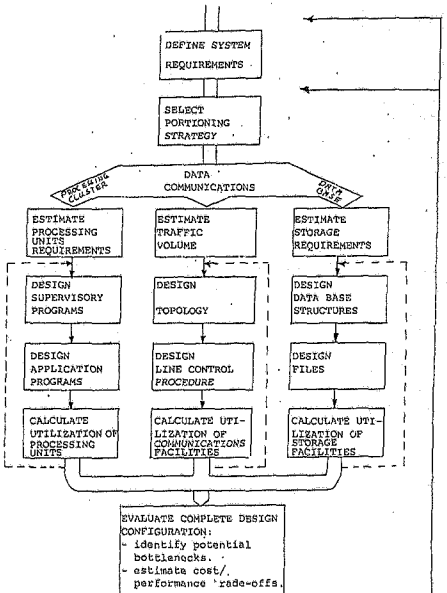


Fig. 2.13

A generalized design sequence for distributed computer systems.

2.7 Fault Tolerance and Recovery

2.7.1 Fault tolerance

Kopetz (15) defines fault tolerance as the detection *of erroneous behaviour and* correcting it dynamically. Prevention is better than cure, i.e. it is better to avoid the occurence of a fault rather than correct it. Every effort is made to eliminate all possible faults before the system becomes operational
e.g. verification of the design,
exhaustive testing
selected hardware components
etc.

2.7.1.1 Failure fault model

A failure is an event and a fault is a state. An event occurs at a certain point in time and a state *lasts for* a given interval. A failure is a fault starting event.

2.7.1.2 Objective of fault tolerant computing

The objective of fault tolerant computing is to detect *faults before they lead to failures.*

2.7.1.2.1 Steps in fault tolerant computing

- fault (error) detection
- fault diagnosis: locate the SRU
- repair or *reconfiguration (if* permanent fault).
- state recovery
- repair of redundant units.

Kopetz (15) goes on to describe briefly each one of these points. Other text books can be consulted for further details on the above mentioned steps.

2.7.2 Recovery

Recovery can be defined as the restoration of the internal state of a component such that it is consistent with its environment.

2.7.2.1 Stateless components

P-stateless: state restoration is not necessary.

M-stateless: initialize the component with the "Static" information structure.

2.7.2.2 State components

There are two approaches to recovery of this type of component:

a) Backward recovery: restore a "previous" state which has been saved at a previous point in time (checkpoint). Modify this state or the environment in such a way that they are consistent again (compensation).

b) Forward recovery: try to find a state which will be consistent with its environment in the near future.

2.7.2.3 Backward recovery in distributed systems

Backward error recovery depends on the provision of recovery points, i.e. a means by which a state of a process can be recorded and later reinstated. Various techniques can be used for obtaining such recovery points.

Reference 23 discusses techniques such as: checkpointing-type mechanisms which involve the foreknowledge of the source from the processes could modify while they are in the atomic action (e.g. all of working storage). Audit trail techniques involving recording all the modifications that are actually made. Recovery cache-type, the recovery block scheme and many others are also mentioned. A major problem with strategies based simply on records of what information flow has, or might have, occurred is that they do not necessarily result in the location of a set of usable recovery points. The problem is illustrated in fig. 2.14. This shows three processes, each of which has three recovery points in addition to that taken at the point of its entry to the shared atomic action. The dotted lines mark occurences of information flow between processes.

Process 1 can only be backed up to its third recovery point. Process 2 can also be backed up to its third point or else process 1 starts interfering with it. Process 3 cannot be backed up without processes 1 and 2 interfering with it.

The search for a usable set of recovery points for the process which is in error, and for other processes which are also affected, is in fact a search for a set of consistent recovery points. Such a set we will term a recovery line. Each process will at any moment have associated with it a (possibly empty) sequence of recovery lines. A recovery line in fact identifies a set of recovery points, each belonging to a different process, such that

a) one of these processes is the process in question;

b) no information flow took place between any pair of processes in the set during the interval spanning the saving of their identified recovery points.

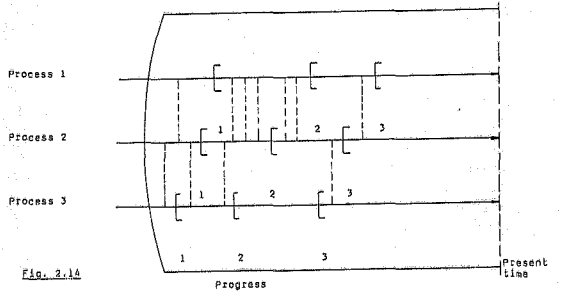


Fig. 2.14

The Domino effect

c) no information flow took place between any process outside the set and any process in the set subsequent to its identified recovery points.

All this leads to the definition of a recovery line as: a line that identifies a set of process states which might have all existed at the same moment, and since which all the processes have been isolated from the rest of the system, so that abandoning the activity which postdates these states is a straight forward task.

2.7.2.3.1 *Limits to Backward recovery*

In a real time system the environment can be considered as a process which cannot be reset to a previous state, e.g. a chemical reaction.

2.7.2.4 Forward recovery in distributed real time systems

The relative simplicity of backward recovery is due to two facts: first, that questions of damage assessment and repair are treated quite seperately from those of how to continue to provide the specified service; and second, that the actual damage assessment takes virtually no account of the nature of the fault involved. In forward error recovery these questions are intermingled, and the technique is to a much greater extent dependant on having identified the fault, or at least all its consequences. Thus generalized mechanisms for backward error recovery are quite feasible. In contrast, *forward error recovery schemes must, it seems, be designed* as integral part of system they serve. Despite this entanglement with all of the other aspects of a system, *forward error recovery techniques* can be quite simple and effective. However this simplicity will be based on the assumed simplicity of the faults and the ensuing damage that they are expected to have to cope with.

Where such assumptions are justified, forward error recovery can be both simple and much more efficient than backward error recovery (after all, backward error recovery involves undoing *everything a system has done since* passing its recovery point, not just the things it did wrongly). However in many cases the simplicity owes much to the fact that the forward error recovery provisions are not even required to achieve complete error recovery.

2.7.2.5 Software design for fault tolerance

a) Distinguish between

i) state component

ii) M-stateless component

iii) P-stateless component

b) Try to reduce the number of state components to a minimum.

c) Analyse the state information of the state components and eliminate that part of *the state information*, which a component can enforce on the environment or which is available in the environment.

d) observe the interfaces around components for error detection (plausibility checks)

e) *use timing checks for error detection.*

CHAPTER 3

COMPONENTS

3.1 Technology and Economics: IC Geometry and Optimum Chip Size

3.1.1 Logic Element Density

The number of logic elements (gates, flipflops) that a manufacturer can lay out on a chip of given area is also obviously of great importance to the system designer, for it ultimately determines his component cost per gate or bit. Logic element density is a complex function of many things, but a useful approximation takes three factors into account.

Suppose

Ac= Area per component ($mils^2$)

Nc= Number of components per gate or per memory bit

S = Stacking factor

$$= \frac{\text{Area actually occupied by gates or bits}}{\text{Total chip area}}$$

$$= \frac{Ac\ Nc \times (\text{Number of gates or bits per chip})}{\text{Total chip area}}$$

All three of these factors have improved over the years. Component area has fallen as manufacturers invented new circuit geometries, and improved the optical systems used to establish component geometry on the chip. Components per gate or bit have fallen as a result of designer ingenuity; and the stacking factor has increased partly as a result of designer experience, but also because proportionally less chip area is wasted on large than on small chips. The table shows roughly how these three factors have changed with time, and figs. 3.1(a) and 3.1(b), (from ref. 18) show the resulting computed circuit densities, in components or gates, and bits per square inch of chip area, respectively.

Geometric Factors* For Integrated Circuit Components

	1962	1966	1970	1974	1978
Component Area A (mils²)					
Bipolar Logic	27	18	8	4	3
Bipolar Memory	27	18	3	3	0.8
MOS Logic			3	1.5	0.7
MOS Memory			3	1.5	0.3
Stacking Factor S	0.10	0.24	0.40	0.40	0.40
Components/Element N					
Per Gate	8	5.5	4	4	3.8
Per Bit—Bipolar	4	4	4	4	4
Per Bit—Static MOS			6	6	6
Per Bit—Dynamic MOS			4	4	2
Elements/in² ** (000)					
Gates—Bipolar	0.412	2.73	20.0	25.0	37.4
Gates—MOS			33.3	66.7	144.7
Bits—Bipolar	0.929	3.75	20.0	50.0	125.0
Bits—Static MOS				44.4	222.2
Bits—Dynamic MOS			33.3	66.7	666.7

*For details on origin of this data, and additional data covering intermediate points, see Ref 3
**Calculated = S x 10⁶/A. N

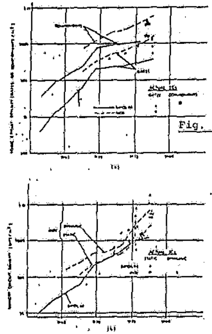


Fig. 3.1 Logic and memory circuit densities. Solid lines show trends in bipolar logic (a) and memory (b) circuit densities, while dashed lines indicate MOS densities as computed from model given in Table. Since early 70s, MOS technology has improved faster than bipolar technology. Points on both curves plot various commercially available ICs, emphasizing fact that individual circuits do not conform to model, which is intended to represent average technology.

Some actual IC data appear on these graphs to emphasize that the model is an approximation and that typical circuits may be quite different for various reasons. Figs. 3.2(a) and 3.2(b) below is a repeat of Montgomery Philster's (18) curve of the number of components and bits on constant-cost chip. Other lines represent data from published papers. Fig. 3.2(a) shows number of components per constant-cost chip. The solid lines which roughly follow the computed points for bipolar and MOS logic. The dotted lines repeat a curve of Noyce's (20). Fig. 3.2(b) shows computed points for number of bits on a constant-cost chip, along with a dotted line representing an equation by Bell (2). Note that the model portrays slower growth rate than those shown in published papers. The rates shown by Noyce (20) and Bell (2) cannot be sustained indefinitely and may already have fallen off.

3.1.2 Optimum chip size

Fig. 3.3 shows how bipolar chip cost has changed with time. Each curve represents chip cost in a given year. Fig. 3.4 provides another look at the same data, and shows that in any year there is an optimum sized chip for which gate costs are minimized. Every year as IC technology improves, the optimum chip gets bigger and the cost per gate smaller. Characteristics of optimum chips appear in figs. 3.3 and 3.4. The chip size which minimizes IC cost is not necessarily the size which minimizes system cost, however, for system costs include the costs of packaging, power and interconnects. Furthermore, the chip which minimizes the system costs is smaller than that which minimizes life cycle costs, which include development and maintenance as well as manufacturing cost. Infact IC manufacturers are constantly looking for designs which incorporate many components and can be sold in large quantities.

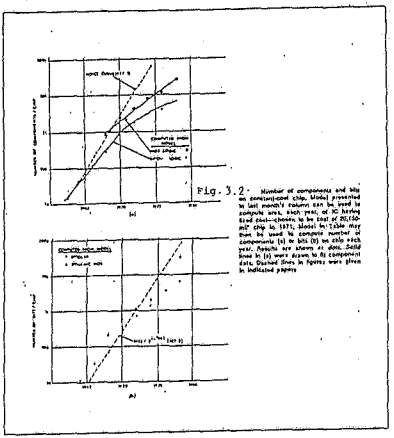


Fig. 3.2 · Number of components and bits on constant-cost chip. Model presented in last monthly returns can be used to compute area, stock-year, of IC having fixed stock—chosen to be that of 20,000 bit chip in 1975. Model in Table may then be used to compute number of components (a) or bits (b) on chip each year. Results are shown as dots. Solid lines in (a) were drawn to fit component data. Dashed lines in figures were given in indicated papers

Memory chips, microprocessors, and programmed logic arrays have emerged as the most important big-chip products to date. But engineers still make use of the far from optimum SSI and MSI circuits for many applications. System designers have found it difficult to make effective use of the potential benefits of IC economics.

3.2 Microprocessors:

With their introduction, microprocessors were hailed as the next generation of digital building blocks. Coupled with nonvolatile memories, they offered a low cost, versatile alternative to minicomputers and hard-wired logic. The inherent flexibility of these devices means that designing microprocessor based products has become largely a matter of simply redesigning system software and reconfiguring I/O, thus allowing the same basic microprocessor to be used in a variety of different applications.

Fulfilling the fixed memory requirement through the use of EPROMS and ROMS has had a direct impact on the versatility of microprocessor systems. Additionally the advent of 5V device operation has significantly reduced power supply requirements and associated costs of design, materials, assembly and reliability. Also, compatibility within families of nonvolatile memories has lessened restraints on device usage. Virtually all MOS memory manufacturers strive for pin compatibility between ROMS and EPROMS of varying densities. This enables the economical use of read only memory for large volume applications and pin compatible erasable programmable read only memory for custom and prototyping applications. Memories of upgradable density make it possible to use the most cost-effective density and allow for changes in program length by providing ease of expansion. These factors are necessary to fully utilize a basic microprocessor in a variety of tasks.

Memory chips, microprocessors, and programmed logic arrays have emerged as the most important big-chip products to date. But engineers still make use of the far from optimum SSI and MSI circuits for many applications. System designers have found it difficult to make effective use of the potential benefits of IC economics.

3.2 Microprocessors:

With their introduction, microprocessors were hailed as the next generation of digital building blocks. Coupled with nonvolatile memories, they offered a low cost, versatile alternative to minicomputers and hard-wired logic. The inherent flexibility of these devices means that designing microprocessor based products has become largely a matter of simply redesigning system software and reconfiguring I/O, thus allowing the same basic microprocessor to be used in a variety of different applications.

Fulfilling the fixed memory requirement through the use of EPROMS and ROMS has had a direct impact on the versatility of microprocessor systems. Additionally the advent of 5V device operation has significantly reduced power supply requirements and associated costs of design, materials, assembly and reliability. Also, compatibility within families of nonvolatile memories has lessened restraints on device usage. Virtually all MOS memory manufacturers strive for pin compatibility between ROMS and EPROMS of varying densities. This enables the economical use of read only memory for large volume applications and pin compatible erasable programmable read only memory for custom and prototyping applications. Memories of upgradable density make it possible to use the most cost-effective density and allow for changes in program length by providing ease of expansion. These factors are necessary to fully utilize a basic microprocessor in a variety of tasks.

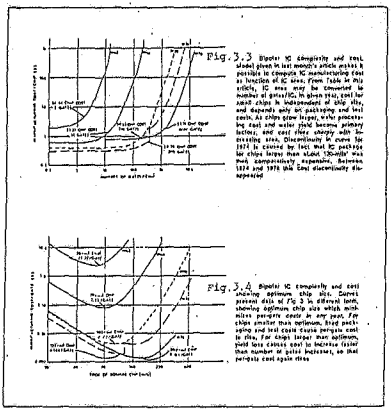


Fig. 3.3 Bipolar IC complexity and cost model given in last month's article makes it possible to compute IC manufacturing cost as function of IC area. From table in this article, IC area may be converted to number of gates/IC. In given trail, cost for small chips is independent of chip size, and depends only on packaging and test costs. As chips grow larger, wafer processing cost and wafer yield become primary factors, and cost rises sharply with increasing area. Discontinuity in curve for 1974 is caused by fact that IC package for chips larger than about 120 mils² was much comparatively expensive. Between 1974 and 1978 this cost discontinuity disappeared

Fig. 3.4 Bipolar IC complexity and cost showing optimum chip size. Correct personal data of Fig 3 in different lines, showing optimum chip size which minimizes per-gate costs in any year. For chips smaller than optimum, fixed packaging and test costs cause per-gate cost to rise. For chips larger than optimum, yield loss causes cost to increase faster than number of gates increases, so that per-gate cost again rises

Fig. 3.5 below shows the cost of microcomputers compared to minicomputers. Microprocessors are becoming cheaper by the day. This figure has been reproduced from Kopetz (15).

3.3 State- and Stateless Components:

If the persistent state of a component is always empty the component is called stateless, otherwise it is a *state component. The distinction between stateless-* and state-components is important from the point of view of recovery. A stateless component is M-stateless (*modifiable*), if its function can be modified without changing the hardware specification. A stateless component is P-stateless (permanent), *if its function can* only be modified by modifying the hardware.

3.4 Failure Rate of Components

The following diagram (fig. 3.6) shows the failure rate *of components with time.* It is a repeat of Kopetz (15) p.2/11.

3.5 *Smallest Replaceable Units (SRUs)*

In order to design distributed systems for *maintainability* and availability (small MTBF and MTTR) the hardware must be designed around easily replaceable SRUs. An SRU may be a *printed circuit board with a collection* of ICs mounted on it. The SRU is normally designed to execute one function. A faulty SRU is simply unplugged from *the system and a non faulty SRU plugged in its place.* The SRU concept makes it easy for the maintenance crew to keep the required spares.

3.5.1 Requirements of SRUs:

It is required that an SRU

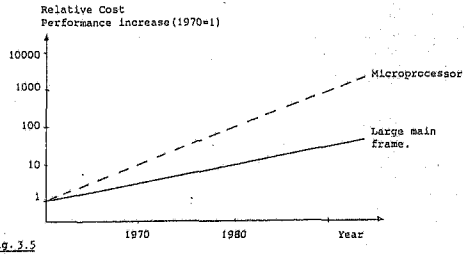


Fig. 3.5

Progress in Microelectronics

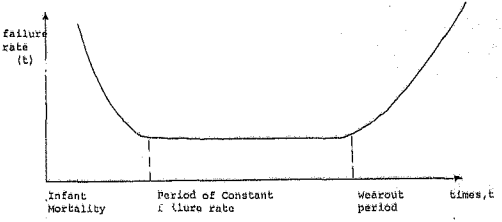


Fig. 3.6

Failure rate of components

- should have simple and observeable interfaces around it, to facilitate testing, e.g. during commissioning and maintenance.

- should be easily tested, i.e. it should be possible to *inject simulated* inputs and observe and study outputs to see if the SRU is faulty.

CHAPTER 4

DCCS REQUIREMENTS OF CASE STUDY

4.1 Brief Description of Case Study
(Watertreatment plant of a Power Station)

The source of all water is rainfall and the properties of raw water will depend on the kind of impurities dissolved by the rain water as it drains away over or through the earth to the point from which the supply to the power station is obtained. Raw water is a subject of very close study and should be analysed carefully to ascertain its suitability for power station use.

Chemically, these waters can be classed as hard or soft, according to the impurities they have picked through contact with the earth. Natural waters also contain certain gases in addition to solid impurities. Most river waters contain mud in suspension and various types of bacteria which could endanger health if used for drinking purposes without treatment.

4.1.1 Water purity

In power station boilers, impurities are undesirable and often dangerous, and steps must be taken to ensure that impurities are eliminated from boiler feed water.

For drinking purposes, water must be clarified and the bacteria eliminated. For most cooling purposes the water can usually be used without treatment. To conserve water, and for other reasons, pre-treatment and partial clarification is becoming accepted practice.

Having discussed the source of water, let us now consider briefly the various purposes for which water is used in a power station.

4.1.2 Various purposes for which water is used in a power station

Power stations obtain their water from natural resources as outlined above. This water is used for:

a) Cooling purposes

b) Supply for the boiler

c) Potable and fire extinguishing water

The water used is clarified and then demineralized so that corrosion and clogging due to crystalization can be minimized.

The system can therefore be divided into three sections namely:

a) Clarification

b) Demineralization

c) Condensate Polishing Plant (CPP).

Under the heading "clarification" we define the
i) Process water treatment plant (Pretreatment plant).

ii) Filtration plant.

Under the heading "Demineralization" we define the boiler feed water make-up system.

Under the heading "CPP" we define the turbine condensate treatment.

There is also a chemical regeneration system; a chemical storage and dosing system; an effluent recovery system to maximize the use of the water; and an effluent concentration/processing system. The latter will not be discussed in this project report.

4.2 Watertreatment Plant or System Requirements

The watertreatment plant of a power station is basically required to:

- produce potable water (for drinking, etc.) for the whole power station.

- produce fire water for extinguishing any fires that may arise in the power station.

- produce demineralized water to be fed into the boiler and used as a means of thermal energy transfer.

- demineralize the boiler-turbine condensate (steam which has been condensed to water at the end of the heat energy transfer process - at the outlet of the turbine) before it is fed back to the boiler.

- clarify raw water before it is used (in the condenser) to condense boiler-feed water before being fed back to the boiler.

4.3 Partitioning Strategy

The DCCS design for this plant will be process orientated for the reasons outlined in chapter 2.4 above.

For the purposes of the application of DCCS techniques to the water treatment plant; this plant will be divided into eight systems namely (see fig. 4.2a)

1. Raw water clarification and chemical dosing (mainly soda ash, alum, lime and polyelectrolyte) - designated system 100.

2. Clarified water storage, filtering and potable water processing - designated system 200.

3. Boiler feed-water demineralization - designated system 300.

4. Chemical storage, dilution/metering and dosing (mainly H_2SO_4 & NaOH) - designated system 400.

5. Condensate Polishing Plant (CPP) - demineralization of the condensate from the boiler-turbine-condenser system - designated system 500.

6. Resin regeneration (cation and anion resin) and storage - designated system 600.

7. Hydrazine and Ammonia storage, dilution/metering and dosing - designated system 700.

8. Demineralized water tankage; effluent drains, recovery and neutralization; and plant air and potable water distribution in the water treatment plant - designated system 800.

The ninth system is the effluent concentration plant which will not be discussed in this project report.

Each system (on fig. 4.2(a)) can be subdivided into more than one subsystem (see fig. 4.1 sheet 1 to 8). System 100, for example, can be subdivided into eleven subsystems namely: subsystem 100, 101, ... 110. Each subsystem is dedicated to a specific task in the system.

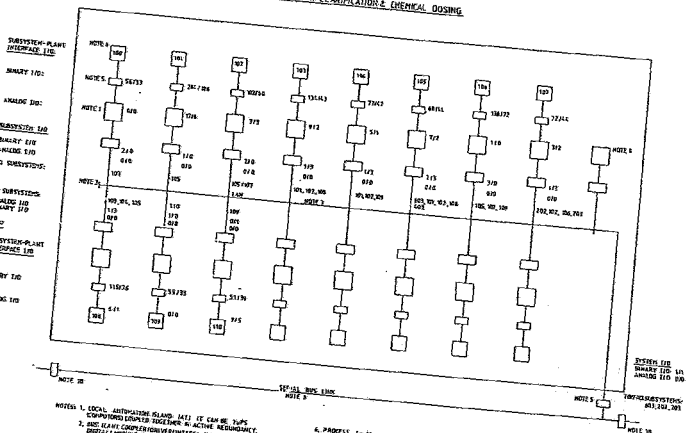
This partitioning is summarised on fig. 4.1 sheets 1 to 8 which shows all the processing clusters involved in the DCCS. Signal flow or characteristics requirements of each cluster are summarised below in tabular form (table 1). Table 1 gives a list of analogn and binary I/O for each system/subsystem for the case study.

	Number of Inputs		From subsystems		Number of Outputs		To subsystems		
System	Binary	Analog	Binary	Analog	Binary	Analog	Binary	Analog	Remarks
100	1	0	803	-	1	0	202 & 203	-	
200	4	0	107,301,402 ESCOM	-	3	0	107,301 ESCOM	-	
300	3	0	800	-	3	0	202	-	
400	4	0	800,105	-	2	0	201,401	-	
500	15	0	600,601,602 ESCOM	-	63	0	600,602,602 ESCOM		
600	63	0	500,501,502	-	15	0	500,501,502		
700	6	0	ESCOM	-	0	0	-	-	No outputs to other systems/ subsystems
800	0	0	-	-	11	0	105,301,404	-	No inputs from other systems/ subsystems

Table 1 Number of inputs/outputs to/from each system/subsystem

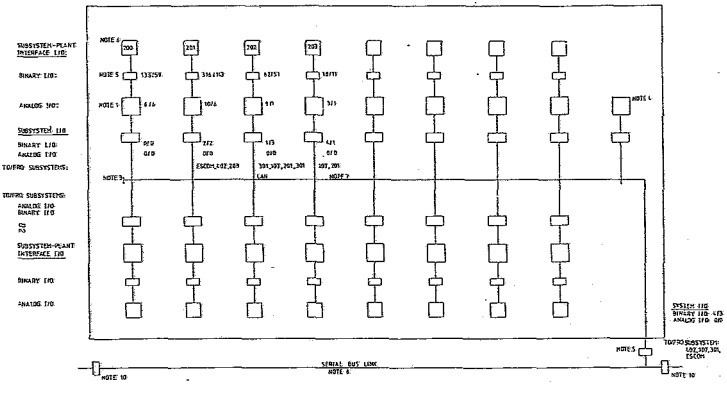
FIG. 4.1 SYSTEM NO 100 SUBSYSTEMS 100 - 110

SYSTEM DESCRIPTION: RAW WATER/EQUALISING COOLING WATER CLARIFICATION & CHEMICAL DOSING



NOTES 1. LOCAL ANNUNCIATOR/ALARM UNIT (IT CAN BE TWIN COMPUTERISED DISPLAY TOGETHER W. ACTIVE REDUNDANCY, DIGITAL/ANALOG).
2. BUS ISOLATOR/CONVERTOR/REPEATER/FAULT ISOLATOR (DIGITAL/ANALOG).
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION/CRT.
5. PROCESS INTERFACE (DIGITAL/ANALOG).
6. PROCESS I - M.
7. LOCAL AREA NETWORK (DIGITAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS CONNECTIONS CAN BE IMPLEMENTED W. ACTIVE REDUNDANCY.
10. BUS COUPLER

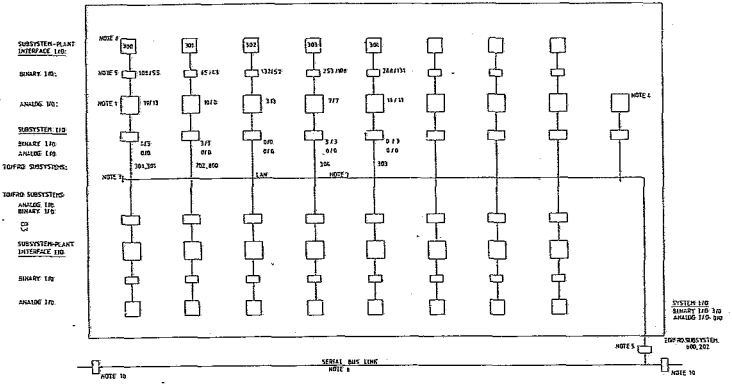
FIG 6.1 SYSTEM NO: 200 SUBSYSTEMS: 200 - 203
SYSTEM DESCRIPTION: FILTRATION, CLARIFIED WATER STORAGE AND CHLORINATION



NOTES: 1. LOCAL AUTOMATION PLANT UNIT (T CAN BE DUPLICATED/COUPLED SOFTWARE IN ACTIVE REDUNDANCY.
2. BUS PLANT COUPLER/DRIVER/INTERFACE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION(S).
5. PROCESS INTERFACE (DIGITAL/ANALOG).
6. PROCESS 1 - 16
7. LOCAL AREA NETWORK (LAN) (DIGITAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS CONNECTIONS/LAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10. BUS COUPLER

SA 2 -

FIG. 6.1 SYSTEM NO. 300 SUBSYSTEMS: 300 – 306.
SYSTEM DESCRIPTION: WATER DEMINERALISATION SYSTEM

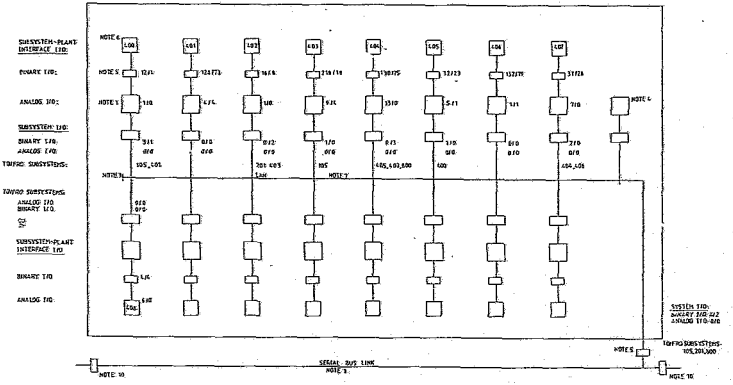


NOTES: 1. LOCAL AUTOMATION ISLAND (AI) IT CAN BE 2xPC COMPUTERS COUPLED TOGETHER IN ACTIVE REDUNDANCY.
2. BUS BASE COUPLER/DRIVER/INTERFACE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION (ES).
5. PROCESS INTERFACE (DIGITAL/ANALOG).
6. PROCESS 1 – 16.
7. LOCAL AREA NETWORK (LAN) DIGITAL/ANALOG.
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS CONNECTIONS/LAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10. BUS COUPLER.

SH. 3*

FIG. 6.1 SYSTEM NO. 600: SUBSYSTEMS: 600–608

SYSTEM DESCRIPTION: CHEMICAL STORAGE, METERING AND INJECTION

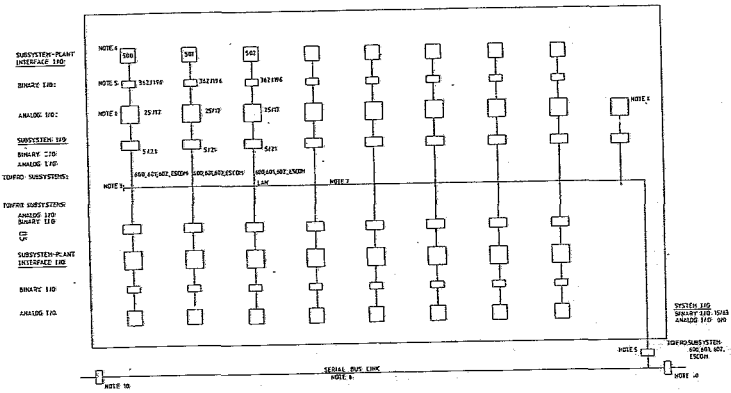


NOTES: 1. LOCAL AUTOMATION ISLAND. EACH IT CAN BE 2xPS
COMPUTERS COUPLED TOGETHER IN ACTIVE REDUNDANCY.
2. BUS CLASS COUPLER/DRIVER/INTERFACE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION(S).
5. PROCESS INTERFACE (DIGITAL/ANALOG).
6. PROCESS 1 – 16.
7. LOCAL AREA HYDRODIC (LAN) (DIGITAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS 1 STATIONS/PROCESS CONNECTIONS LAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10. BUS COUPLER

SH 6-

FIG. 6.3 SYSTEM NO: 500 SUBSYSTEMS: 500 - 502

SYSTEM DESCRIPTION: CONDENSATE POLISHING SYSTEM



NOTES: 1. LOCAL AUTOMATION CLAMP RACK IT CAN BE 2xPS (COMPUTORS) COUPLER TOGETHER IN ACTIVE REDUNDANCY.

2. BUS (LAN) COUPLER/DRIVER/INTERFACE (ANALOG/ANALOG DIGITAL) MODULE.

3. BUS TERMINATOR.

4. LOCAL OPERATING STATION (OS).

5. PROCESS INTERFACE (DIGITAL/ANALOG).

6. PROCESS 1 - 16

7. LOCAL AREA NETWORK (LAN) (DIGITAL/ANALOG).

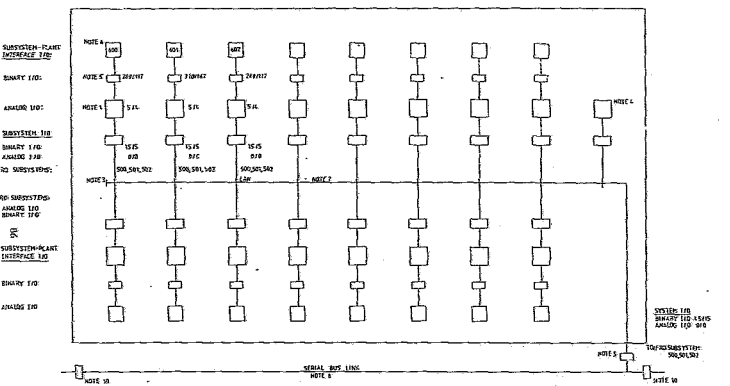
8. SERIAL BUS LINK (DIGITAL/ANALOG).

9. ALL 600 SYSTEMS/PROCESS CONNECTIONS/LAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.

10. BUS COUPLER.

SH. 5-

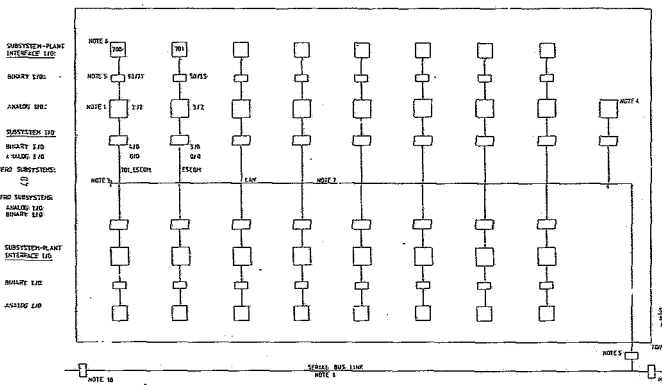
FIG. 4.1 SYSTEM NO. 600 SUBSYSTEMS: 600-602
SYSTEM DESCRIPTION: REGENERATION SYSTEM



NOTES: 1. LOCAL AUTOMATION ISLAND (ALL ITS CAN BE 2xPS (COMPUTERS) COUPLED TOGETHER IN ACTIVE REDUNDANCY.
2. BUS BOARD (COUPLER/DRIVER/MAINTENANCE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION/SL
5. PROCESS INTERFACE (DIGITAL/ANALOG)
6. PROCESS 1 - 16
7. LOCAL AREA NETWORK (CAMAC/SERIAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS COMPUTERS/PLAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10. BUS COUPLER

SK 4-

FIG. 6.1 SYSTEM NO.700 SUBSYSTEMS: 700 - 701.
SYSTEM DESCRIPTION: AMMONIA AND HYDRAZINE STORAGE, DILUTION AND INJECTION.

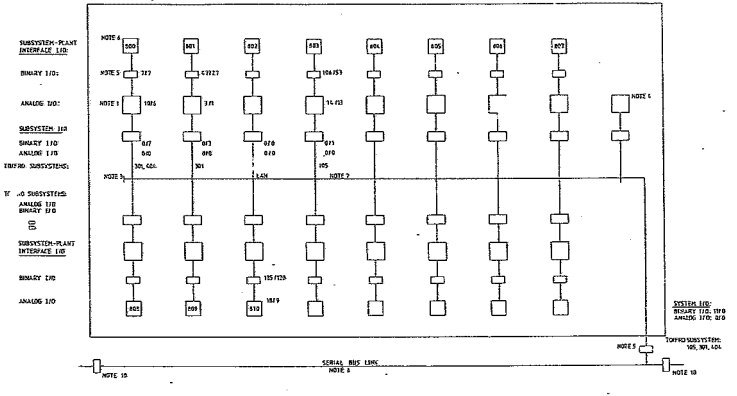


NOTES: 1. LOCAL AUTOMATION ISLAND: EACH IS CAN BE 2 uPS COMPUTERS COUPLED TOGETHER IN ACTIVE REDUNDANCY.
2. BUS ISLAND COUPLER/DRIVER/INTERFACE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION(S).
5. PROCESS INTERFACE (DIGITAL/ANALOG).
6. PROCESS 1 - 16
7. LOCAL AREA NETWORK (LAN) (DIGITAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS CONNECTIONS CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10. BUS COUPLER

SHT 3-

FIG. 4.1 SYSTEM NO. 800 SUBSYSTEMS: 800 – 810

SYSTEM DESCRIPTION: DEMIN WATER TANKAGE, EFFLUENT NEUTRALISATION PLANT AIR & POTABLE WATER DISTR



NOTES: 1. LOCAL AUTOMATION ISLAND UNIT IF CAN BE DUPS. COMPUTERS COUPLED TOGETHER IN ACTIVE REDUNDANCY.
2. BUS ISLAND COUPLER/DRIVER/INTERFACE (ANALOG AND/OR DIGITAL) MODULE.
3. BUS TERMINATOR.
4. LOCAL OPERATING STATION(S).
5. PROCESS INTERFACE (DIGITAL/ANALOG)
6. PROCESS 1 - 16
7. LOCAL AREA NETWORK (LAN) DIGITAL/ANALOG).
8. SERIAL BUS LINK (DIGITAL/ANALOG).
9. ALL BUS SYSTEMS/PROCESS CONNECTIONS/LAN CAN BE IMPLEMENTED IN ACTIVE REDUNDANCY.
10 BUS COUPLER

SM 8 -

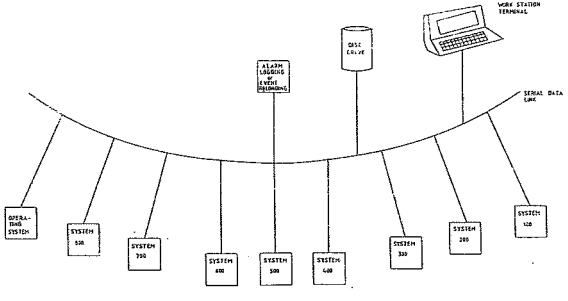


FIG. 4.2(a) PROPOSED DCCS BLOCK DIAGRAM OF CASE STUDY

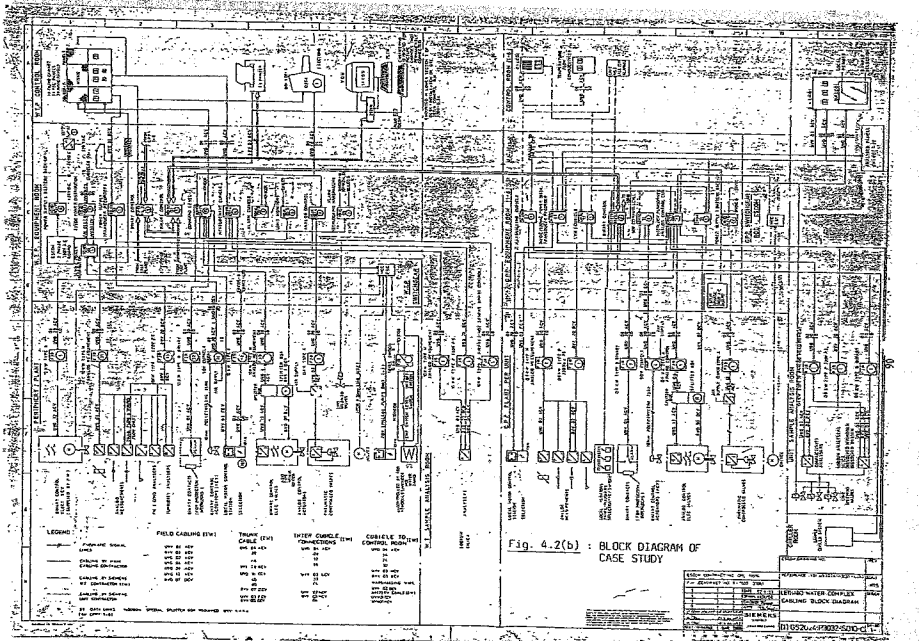


Fig. 4.2(b) : BLOCK DIAGRAM OF CASE STUDY

4.4 Processing clusters

4.4.1 Block diagram of case study

Fig. 4.2(b) above gives the block diagram of the case study in the existing design. Fig. 4.2(a) above is the proposed block diagram; and is much simpler than the former. This simplified block diagram depicts one of the benefits to be achieved by using a distributed computer control system for the control system of this plant.

Fig. 4.1 (sheets 1 to 8) shows subsystems of each major system (e.g. subsystem 103 in system 100) and the automation islands associated with each subsystem. The number of inputs and outputs of each subsystem are also given.

It will be noticed that the number of I/Os handled by the process interface module (note 5) are more than those handled by the LAN interface module (note 2). The number of I/Os handled by the latter are more than those handled by the serial bus coupler (note 10).

Generally, data communication rates are high between the partial process (or automation islands) and the partial process controller; the data rate is low between partial process controllers but is even lower between major systems.

4.4.2 Processing units requirements:

Each processing unit (or automation island) will be required to handle a maximum I/O of 362/196 (on the process side) according to the information given on fig. 4.1 sheets 1 to 8. The automation islands/ processing units will therefore be designed for an I/O of 500/500 in order to allow for any expansion in each system. Each processing unit is dedicated to a subsystem and thus more processing units will be required whenever new partial processes are added.

The I/O requirements of each whole system are minimal as shown on fig. 4.1 sheet 1 to 8. System 100 I/O for example, is 1/1 for Binary signals and O/O for analog signals. System with maximum input requirements is system 600; and I = 45. System 600 has maximum outputs = 63. The standard interface between systems can be designed to handle an I/O of say 100/100.

Additional requirements of the processing units would be analysed thoroughly, and thereafter design supervisory programs, application programs, calculate utilization of process units, etc. as shown on fig. 4.3 below.

The "data communication" and "data base" systems would then be designed step by step (for each) as shown on the same fig. The design of the DCCS would be complete once this general design sequence is complete. It is proposed that this design sequence be adopted and used for the case study. Some of the aspects that have to be dealt with during the design process (as given by fig. 4.3) are given below from chapter 4.4.3 to the end of chapter 4.

4.4.3 Signalling system

Signals represent information on extra ordinary or unfavourable events in the course of a process. They are unexpected items of information which cannot be recognised as the obvious necessary consequence of a previous occurrence. There are two basic forms of signals in a plant:

i) process signals, status signals and

ii) control system internal alarms.

Process signals and status signals communicate an exact picture of the process to the operator (control room personnel). These signals include limit values of measurement values (pressure, temperature, flow, etc.) and status signals (status feedback such as ON/OFF, OPEN/CLOSED, MANUAL/AUTOMATIC, present criteria).

Control system internal alarms come from the automation equipment; i.e. the control system components. They are faults from the automation equipment.

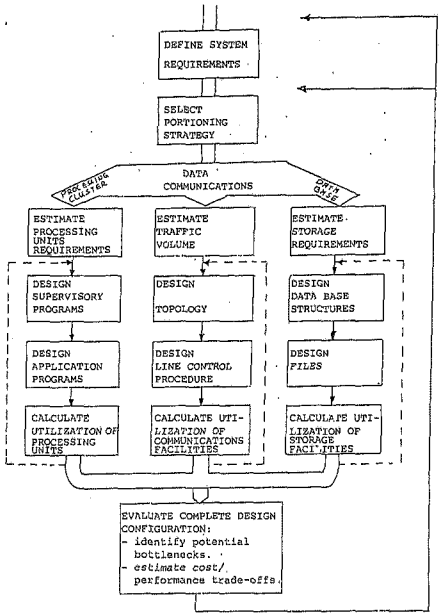


Fig. 4.1

A generalized design sequence for distributed computer systems.

Control system components include sensors, transmitters, transducers, actuators, switching units, power supply units, electronic modules and connecting cables. In addition, the functional interaction of these components is monitored for faults. Qualitative distinguishing features are:

i) direct signalling (without bus subsystem)

ii) central signalling (with bus subsystem).

With direct signalling, the most important feature is the complete independence of the system without it being connected to the [illegible]em. An automation subsystem is then able to d[illegible]lay the process and control system signals a[illegible] it. The process signals can be recognised by the operator on the monitor or the manual automatic control station.

All process signals, status signals and control system internal alarms will be grouped together within each local processor and transmitted to the monitor or data logger via BUS 1.

The scope of the alarms will cover all vital signals or alarms associated with the operation and monitoring of the Water Treatment Plant.

4.4.4 Operating system (OS)

The operating system consists of colour monitors and light pens (full graphics with curve display), etc. Serial operator communication and monitoring is required, so that only subsets of the total information and associated intervention facilities are presented to the operator at one time (monitor with display call-up).

If pure serial operation by the monitor is not appropriate for reasons of availability or reliability, manual/automatic control stations, acting in parallel, can be used *in addition*. *Parallel and serial* operator communication systems can be optimally combined for the particular automation problem. Operator communication units with process communication keyboard and a structuring keyboard and a monitor may be used for direct operator communication and structuring of an automation *subsystem*.

The use of monitors enables better organization, clear display of information and thus a reduction in stress for the operators and a better handling of faults. The space requirements in the control room are considerably reduced by the use of monitors.

A characteristic of this process communication and monitoring is the hierarchical organization of the information. The operator can select different levels in the hierarchy, starting with an overall view of the process (coarse information) down to the display of detailed images (e.g. a single control loop).

Only those operating possibilities which are permissible for the partial process selected are offered for operator guidance. The probability of maloperations is thus greatly reduced. The data representation on the monitor is based on years of *experience in the field of ergonomics, especially* anthropotechnics.

4.4.5 Operator communication unit

Operator communication and monitoring of the individual automation subsystems, as well as the output of signals and logs, are carried out using the operator communication unit, a combination of interface module, process communication keyboard and a monitor.

The keys on the process communication keyboard are automatically lettered according to the particular function and the possible operations thus indicated to the operator. The displays for operator communication and monitoring have a hierarchical construction (area display, group display, loop display).

4.4.6 Subsystems within each major system

The automation subsystems have integrated operation (operating program, block lists, operable blocks, etc). The automation subsystems can be operated in stand-alone mode or in conjuction with other subsystems to which they are coupled through the bus system.

The hardware of the automation subsystem consists of a basic unit and one or several extension units. The modules for power supply, the interfaces, the CPU and the memory are accomodated in the central area of the basic unit. Spare locations of the basic unit and in the extension units are available for input/output modules. These modules establish the connection between the automation subsystem and the process. The process signals (analog, binary) are acquired in the input/output modules and prepared for processing in the CPU. The modules also contain output functions for positioning instructions provided by the CPU (e.g. to valves, etc.).

4.4.7 Software

The CPU is microprogrammed. The EPROM's contain the invariable parts of the function blocks and the programs for structuring, process communication and monitoring.

The function blocks are complete program sections (firmware) such as closed-loop blocks and sub-group open-loop control blocks, multipliers, logic elements (AND, OR, etc.) which are processed in multiplex.

The keys on the process communication keyboard are automatically lettered according to the particular function and the possible operations thus indicated to the operator. The displays for operator communication and monitoring have a hierarchical construction (area display, group display, loop display).

4.4.6 Subsystems within each major system

The automation subsystems have integrated operation (operating program, block lists, operable blocks, etc). The automation subsystems can be operated in stand-alone mode or in conjuction with other subsystems to which they are coupled through the bus system.

The hardware of the automation subsystem consists of a basic unit and one or several extension units. The modules for power supply, the interfaces, the CPU and the memory are accomodated in the central area of the basic unit. Spare locations of the basic unit and in the extension units are available for input/output modules. These modules establish the connection between the automation subsystem and the process. The process signals (analog, binary) are acquired in the input/output modules and prepared for processing in the CPU. The modules also contain output functions for positioning instructions provided by the CPU (e.g. to valves, etc.).

4.4.7 Software

The CPU is microprogrammed. The EPROM's contain the invariable parts of the function blocks and the programs for structuring, process communication and monitoring.

The function blocks are complete program sections (firmware) such as closed-loop blocks and sub-group open-loop control blocks, multipliers, logic elements (AND, OR, etc.) which are processed in multiplex.

A language would have to be developed to program the subsystems. The function blocks are interconnected or parameterized by plant-specific structuring data, i.e. input of structuring instructions. The structuring data are stored in a RAM module. To increase reliability and availability, the RAM modules are provided with a battery back-up to prevent data loss in the event of power failure.

Structuring can be carried out using monitors and structuring keyboards (alphanumeric keyboards). An input/output typewriter and a mini floppy disk unit can be used as supplementary devices. Modules with structuring and operating data for every function block installed by the user (parameters, limit values, interconnections, actual measured values, etc.) can be called up on a monitor.

Updating of the documentation is carried out automatically by back-translation from the original data contained as bit patterns in the system. Thus there are no discrepancies between the actual processing and its documentation (real-time documentation). There are function blocks (e.g. controllers and measured-value monitoring blocks) which enable a standardized display on the monitor for process communication and monitoring in addition to the processing function.

Direct process communication is possible using standard input/output devices such as a process communication keyboard with relevant key lettering and a monitor (operator communication unit).

The automation subsystem functions can be summarized as follows:

A. Function blocks for signal processing

1) Signal status display
2) Signal status log
3) Signal sequence display
4) Signal sequence log
5) Control system *internal alarms*

B. Free *function blocks*
1) *Open-loop* function blocks
2) General functions
3) Graphic displays
4) Logging

C. Standard function blocks (extension)

D. Multiple programs
1) Monitoring
2) Closed-loop control
3) Open-loop control
4) Arithmetic operation
5) Logic operations
6) Operator communication
7) Displays
8) Logging
9) *Coupling*
10) *Signalling*
11) Alarm processing

E. Structuring instructions
Basic configurations
1) For control functions
2) For general functions
3) For graphic displays
4) For logging

4.4.8 Communication system

Data is exchanged between individual systems or subsystems via the bus system *BUS 1*. *The main requirement* here is a coupling between several automation subsystems of one and the same plant for the transmission of control information as well as a coupling to the subsystems provided for operator communication, signalling and logging.

This bus system consists of a local bus and a remote bus, for *the transmission of data*. *Bus converters* are used as links between the local and remote buses; and bus couplers are used for linking autonomous buses. *With* these *facilities* data transmission can be designed optimally according to the application.

4.4.8.1 Remote (serial) bus system

Information of various types, compiled into messages, is transmitted between the individual *subsystems via the bus* system in serial mode. The messages are protected and are acknowledged by the receiver if transmitted without errors. The message is transmitted a second time if a collision has been sensed. The data paths can also be *designed with redundancy* where high requirements for availability of data transmission exist.

Each subsystem is coupled to the bus (local) system via the autonomous bus interface module. This module decouples the exchange of data, in time and function, between its subsystem and the bus system. Every bus interface module *can control* the exchange of data, i.e. it can take control of the bus. The capability to control a bus (master function) is random and *any bus interface module* within each autonomous bus system can take control of the bus. This communication protocol is referred to as CSMA /CD (carrier sense multiple access/collision detection) and is a very powerful protocol.

The local and remote buses are connected together through a converter or bus coupler. Data fragments resulting from data collisions can be limited (by the bus decouplers) to one bus only. Thus, a higher availability than with a central bus control is achieved in this way because new data transfer can be automatically set into operation in the other sections of the bus if other parts have data fragments.

The subject of CSMA/CD protocol will not be discussed here, because this information can be found in a lot of papers, e.g. the paper by Shock, Dalal and Redell (24).

4.4.8.1.1 Data structure

Messages transmitted between stations/local automation islands conform to a data structure, the technical details of which are given in appendix 1.

4.4.8.2 Local bus system

The first major step towards general compatibility in electronic instrumentation for system use came in 1975 with the publication of the IEEE 488 - 1975 standard that defined an interface and communication bus for programmable instruments. This bus is commonly called the GPIB - the *General Purpose Interface Bus*.

In 1978, the standard was further refined (*IEEE 488-1978*), defining an interface system that has become a widely accepted instrument industry standard. Traditionally, the bus has had the image of being devoted to the area of instrumentation, control and data acquisition.

It has allowed engineers to concentrate on solving their measurement problems instead of on computer instrument interfacing. With GPIB compatibility, measurement devices can be chosen off-the-shelf and simply cabled with standard bus cables in either a linear or star configuration.

This is the local bus system that is proposed for the Water Treatment Plant. This standard is discussed in many papers and will, therefore, not be discussed any further.

4.4.9 Requirements of the communication system

4.4.9.1 Serial bus system

1. The serial bus system shall have a distributed structure. There is no equipment with central control tasks within the bus system. Each bus interface module can control and monitor the data flow on the bus (master function). This function is not tied to any one interface module and can therefore be acquired by another bus interface module at any time. High availability of the transmission equipment is achieved in this manner.

2. The communication system will have a high extensible transport capacity.

3. The transport delay induced by the communication system must have short and a guaranteed maximum value.

4. The reliability of the communication system must be high: one failing link should not stop communication between other systems.

5. Mechanisms have to be provided in the communication system that allow an uninterrupted monitoring of all resources (computers and links) and the dissemination of the resulting status information.

4.4.9.1.1 Inductive bus converter

A bus converter is needed to connect individual participants or a local bus with several participants to a remote bus. It performs a continuous signal conversion between the local bus and the remote bus or vice versa without intermediate storage.

A coaxial cable or fibre optic link may be used as the remote bus depending on the data rate (coaxial cable for low rates and fibre optic link for high data rates).

4.4.9.1.2 Bus coupler

For the exchange of messages between bus systems working independently. The coupler accepts a message from bus 1 for a participant on bus 2, stores it and transmits it on bus 2 at the earliest opportunity and vice versa.

4.4.9.1.2.1 Parallel/redundant bus couplers

Two bus couplers can be installed between two autonomous remote bus systems. A differentiation must be made between two applications:

4.4.9.1.2.1.1 Parallel bus couplers

If the coupling capacity of a bus coupler is insufficient, the transmitting capacity can be increased by approximately 100% using a second bus coupler. This parallel connection of bus couplers leads to a limitation in the capacity should a bus couple fail (this is not a redundant connection).

Fortunately there are also *special conditions in local* area networks that allow for these extensive requirements to be realized on an economical basis:

i) High capacity lines (e.g. Optical fibre links) may be installed : Line capacities are not a limiting factor.

ii) The transmission error rates are very low - much lower than in the public telephone network.

iii) Propagation delays are negligible.

4.4.10 Characteristics of the communication (bus) system

4.4.10.1 Application

- Exchange of information between the individual participants of the distributed computer process-control *system*.

- Sequential transmission of information (messages) in digital form.

- Connection of external systems through a suitable interface module.

- A bus system with facilities for participants located:
  - far apart in the plant area (remote bus and/or)
  - locally concentrated (local bus)

Remote bus as a connection between distributed DCCS over distances up to 4km.

Local bus as a connection between participants:
- within a cabinet or a group of cabinets over distances less or equal to 20m or -

- over distances up to 100m (with an additional electrical isolation module per participant).

4.4.10.2 Design

The bus system consists of a local and a remote bus. The distinction between the local and remote ranges is significant for transmission and functional reasons.

Transmission modes using inexpensive transmitter and receiver equipment are used in the local range. The number of line signals is not as drastically limited as in the remote range. Redundancy and electrical decoupling do not become too expensive.

4.4.10.2.1 Local range

The data are transmitted sequentially on data lines. Clock, synchronization and control signals are transmitted on signal lines in the local range. Distances of up to 20m are covered with an unbalanced and electrically coupled connection between participants (for use within cabinets and groups of cabinets).

Distances of up to 100m (line length) can be covered with electrically isolated, balanced transmission.

4.4.10.2.2 Remote range

Transmission takes place via a coaxial cable over distances of up to 4km (line length) for data rates in hundreds of thousands of bits. If the data rate is in the range 1M bit - 10M bit a fibre optic link is preferred. The signals are self-synchronizing bipolar square pulses. The coupling to the remote bus is non-reactive. Remote and local buses can be coupled by bus converters.

In this way, groups of automation systems (AS) or operating systems (OS) located in one area and connected by a local bus can exchange data with more distant systems. The complete bus system carries the same data.

Two buses are connected via a bus coupler so that only the data intended for the other bus is transferred. The bus coupler connects two autonomous bus systems. Using these two coupling devices, the requirements imposed on the distribution of data from the automation system in a plant are met.

The following modules are very important for the realization of a "good" and reliable bus system:

4.4.10.2.3 Local bus interface modules

For connection of individual participants to the DCCS bus system. The local bus interface modules are plugged into the basic units of the automation system, operating system and process computers and have an output for a redundant 20-m local bus.

4.4.10.2.4 Electrical isolation module

To extend the 20-m local bus (one isolation module needed per participant).

4.4.10.2.5 Redundant bus couplers

If the coupling capacity of one bus coupler is sufficient, a redundant link can be planned using a second bus coupler. The second bus coupler automatically assumes the coupling function if the first bus coupler fails. This is a true redundant connection.

The technical details of each one of these items will not be discussed because these details depend on the application on hand.

4.4.10.3 Comments on coupling software

The coupling software shall support the carrier sense multiple access/collision detection (CSMA/CD) with a random delay communication protocol. Nothing more than this will be said about CSMA/CD as mentioned before.

20-m local bus

Application:

- very small plants
- up to 4 participants
- up to 20m bus cable
- redundancy as standard.

100-m local bus

Application:

- small plants
- up to 10 participants possible
- distance of 100m between first and last local bus interface module.
- no long term plans for plant expansion.

20-m local bus/4-km remote bus

Application:

- large plants
- up to 100 participants
- up to 4km remote bus cable.

Single participant on the 4km bus

- medium sized plants
- single participation on the remote bus (long distance between individual participants).

- maximum peripheral configuration of the connected automation system.
- up to 32 participants possible
- up to 4km remote bus cable.

4.5 Architecture

The proposed control system will be distributed in three dimensions as shown on fig. 2.1 above.

1. Distributed control

2. Distributed hardware

3. Distributed data

Enslow (1978) gives a detailed discussion of this distribution concept. This concept is also discussed briefly in chapter 2.1 above. The communication system used to support this structure is distributed; and the communication protocol proposed is carrier sense multiple access with collision detection (CSMA/CD) as discussed above.

This subject (architecture) will not be discussed extensively because further details can be obtained in Enslow (1978).

4.6 Fault Tolerance and Recovery

4.6.1 Fault tolerance

Erroneous behaviour of the system shall be detected and corrected dynamically. The techniques/methods that may be used to do this are discussed in chapter 2 above. When the system does ultimately fail, it shall go to a fail safe mode.

4.6.2 Recovery

This system shall be provided with a series of consistent recovery points all of which will be the end of a given "process".

Backward and forward recovery shall be designed into the real time distributed computer control system.

This subject is discussed extensively in chapter 2.7 above and will therefore not be dealt with in great detail in this chapter.

4.7 Reliability and Maintainability

The water treatment plant of a thermal power station is one of the key plants that should work properly all the time if the problems outlined in chapter 1 above have to be avoided. It is therefore important that the reliability of such a control system be fairly high.

The control system should be easily maintainable because a power station is a labour intensive enterprise and therefore require standard hardware/software modules to replace and/or expand the system. The maintenance tasks should be simplified such that they can be done by lowly trained personnel.

4.7.1 Reliability

1. Additional individual manual control stations will be provided.

2. A local operating system will be provided.

4.6.2 Recovery

This system shall be provided with a series of consistent recovery points all of which will be the end of a given "process".

Backward and forward recovery shall be designed into this real time distributed computer control system.

This subject is discussed extensively in chapter 2.7 above and will therefore not be dealt with in great detail in this chapter.

4.7 Reliability and Maintainability

The water treatment plant of a thermal power station is one of the key plants that should work properly all the time if the problems outlined in chapter 1 above have to be avoided. It is therefore important that the reliability of such a control system be fairly high.

The control system should be easily maintainable because a power station is a labour intensive enterprise and therefore require standard hardware/software modules to repair and/or expand the system. The maintenance tasks should be simplified such that they can be done by lowly trained personnel.

4.7.1 Reliability

1. Additional individual manual control stations will be provided.

2. A local operating system will be provided.

3. Redundancy may be built into the system as follows:
   a) Redundant communication system

   b) Redundant devices connected in active redundancy.

   c) etc.

Sources of system unreliability are outlined in chapter 2.2.3 above.

4.7.2 Maintainability

Standard SRUs will be used to facilitate the dynamic replacement of faulty units in the system. The subject of SRUs is discussed in chapter 3 above.

The software shall be structured such that it is easily maintainable over the lifetime of the system. This subject is discussed briefly in chapter 2.2.2.2 above.

4.8 Back-up

This subject will not be treated, but if back-up is required it can be achieved by connecting two automation islands/partial processors in active redundancy to each partial process. This will mean that all output lines from the partial process will have to be duplicated in order to supply information to each automation island. The additional costs of implementing a back-up system in this case, would be very high (see also chapter 2.3.4). This is an extensive subject in its own right and will not be discussed further.

CHAPTER 5

RECOMMENDATIONS AND CONCLUSION

5.1 Recommendations

Distributed Computer Control Systems (DCCS) have a lot of advantages over conventional centralized control methods and should therefore be used more in new process control plants.

From the point of view of maintenance, reliability, safety, recovery, etc., there are definately many benefits to be gained. For economic reasons, the automation of technical processes still needs, and will also need in the future, DCCS expertise. Means must be available for DCCS with which processes of varying automation degrees and levels can be controlled. In addition, adequate process control must be ensured in the event of failure of automatic equipment.

The process data is available in digital form in modern process control systems. Operator communication and monitoring systems with CRT monitors can therefore be connected directly to the programmable digital automation system(s). Operator interventions are input as a character string via a keyboard. Thus new means are available for the improvement of control room design/layout.

5.1.1 What improvements are provided by these new methods?

1. In the design of control stations for large plants with predominantly parallel communication systems, there is always a problem in arranging the many information units and controls in such a way that the operator retains the necessary overall view.

The large numbers of communication units leads to very extensive panels and desks. Seen from the operator's work station (central supervisory position), there are problems of visibility and legibility of information in more distant sections of the control station.

If interventions are necessary, the operator may have to move from the supervisory position to the place of action or may have to call on a second operator. It is possible to carry out the monitoring and operator communication functions at the same place using video display systems since both functions are combined in these systems.

Since operator communication is only necessary to a small extent in the substantially automated process plant, the serial representation of communication possibilities through monitors is adequate.

Since operation with monitors always takes place within groupings of standardized displays which correspond to the arrangement of conventional control and display elements in a control panel section, the necessary parallel relationship is preserved. The use of monitors for operator communication and monitoring enables the clear arrangement of control stations for the operator while retaining all facilities for individual actuation.

If the information for monitoring and clarification of faults has to be highly detailed, this necessitates the arrangement of many communication elements (e.g. indicators, displays etc.) when using conventional means, with a corresponding space requirement.

A highly detailed view of the whole plant is not required continuously by the operator, however, but only in particular cases (stopping and starting of plant sections, fault in a plant section, etc.).

Video display units provide the facility to increase the degree of detail of the information step by step by changing from overview displays to detailed display.

2. A rigid arrangement of the communication elements for different operational cases is always a compromise. With the video display technique, however, the grouping of information and communication facilities can be changed for different operational cases. This brings the advantage that the operator no longer has to find and combine the communication elements necessary for the particular case (operator guidance).

3. Flow diagrams or mimics with display of the variable states provide an adequate image of the plant. The operator can rapidly obtain an overall view of the current functional state. However, extensive flow diagrams in desks and panels require large areas, while operating controls can only be arranged within easy reach. On the other hand, video display units are particularly suitable for the display of flow diagrams with superimposed displays for measured values, switching state of units or position of valves etc. Large displays are possible which can be moved across the screen in rolling map mode using a joystick.

Interventions in these freely designable displays can also be carried out using a light pen and superimposed key field.

4. Line and multipoint recorders are used to record the variation of process variables with time. The charts serve for documentation of the operating sequence for the plant management and also assist the operator in following the trend of measured values in order to decide on any necessary actions. Since the recorders mostly have to be arranged in panels (space requirement), and thus mainly at some distance from the central supervisory position, it becomes difficult to recognize the shape of the curve, the measuring point identification or the unit.

The display of curves on monitors, however, can be examined at a favourable distance at the supervisory position. Bar displays can also be used for measured values instead of curves.

The technical devices and [illegible] t in the DCCS are adapted to this requiremer[illegible]s, namely:

a) direct operator communication and monitoring and

b) central operator communication and monitoring.

5.2 Conclusion

Distributed computer control systems are a new class of organizations and operations that exhibit a high degree of distribution in all dimensions, as well as a high degree of co-operative autonomy in their overall operation and integration. Numerous systems have been designed that meet one or another of the criteria in the definition of distributed systems, however, a large number of research issues must still be solved before we can implement a true and complete distributed computer control system.

Already the implementation of the first generation of DCCS has shown substantial advantages for the user and the vendor.

By providing the process control operator with processed plant information, which is more relevant to his decision-making and by automating routine tasks it has been possible to improve the efficiency of the plant operation.

For the vendor, the decomposition of a monolithic control system into a set of standardized modules (SRUs) has considerably reduced engineering overhead.

The prevailing interest in DCCS is justified by the requirements for reliability and flexibility. But the DCCS has its own problems of unreliability and inflexibility. These problems are similar to those plaguing all computer design but made even more serious when functions are distributed in different physical boxes (SRUs).

The next generation of DCCS will bring further improvements in the efficiency of plant operation by integrating the process control system of an organization. In order to realize such an integration, for the water treatment plant of a thermal power station, a set of design principles for integrated DCCS has been presented together with a set of the open problems, which have to be solved.

Already the implementation of the first generation of DCCS has shown substantial advantages for the user and the vendor.

By providing the process control operator with processed plant information, which is more relevant to his decision-making and by automating routine tasks it has been possible to improve the efficiency of the plant operation.

For the vendor, the decomposition of a monolithic control system into a set of standardized modules (SRUs) has considerably reduced engineering overhead.

The prevailing interest in DCCS is justified by the requirements for reliability and flexibility. But the DCCS has its own problems of unreliability and inflexibility. These problems are similar to those plaguing all computer design but made even more serious when functions are distributed in different physical boxes (SRUs).

The next generation of DCCS will bring further improvements in the efficiency of plant operation by integrating the process control system of an organization. In order to realize such an integration, for the water treatment plant of a thermal power station, a set of design principles for integrated DCCS has been presented together with a set of the open problems, which have to be solved.

APPENDIX 1

DATA STRUCTURE - TECHNICAL DETAILS

A1.1 Packet Format

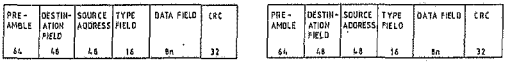


Fig. A1.1 DATA STRUCTURE

Stations/local automation islands must be able to transmit and receive packets on the common coaxial cable/fibre optic link (Bus system) with the indicated format and spacing. Each packet should be viewed as a sequence of 8-bit bytes; the least significant bit of each byte (starting with the preamble) is transmitted first.

Maximum Packet Size: 1526 bytes (8 byte preamble + 14 byte header +1500 data bytes + 4 byte CRC).

Minimum Packet Size: 72 bytes (8 byte preamble + 14 byte header + 46 data bytes + 4 byte CRC).

Preamble : This 64-bit synchronization pattern contains alternating 1's and 0's, ending with 2 consecutive 1's. It is 1010101010101010101010101010101010101010101010101010101010101011.

Destination Address: This 48-bit field specifies the station(s) to which the packet is being transmitted. Each station examines this field to determine whether it should accept the packet.

The first bit transmitted indicates the type of address. If it is a '0', the field contains the unique address of the one destination station. If it is a '1', the field specifies a logical group of recipients; a special case is the broadcast (all stations) address, which is all 1's.

Source address: This 48-bit field contains the unique address of the station that is transmitting the packet.

Type field: This 16-bit field is used to identify the higher level protocol type associated with the packet. It determines how the data field is interpreted.

Data field: This field contains an integral number of bytes ranging from 46 to 1500. The minimum ensures that valid packets will be distinguished from collision fragments.

Packet check sequence: The 32-bit field contains a redundancy check (CRC) code, defined by the generating polynomial:

$G(x)=x^{32}+x^{26}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x^{1}+1$

The CRC covers the address (destination/source), type, and data fields. The first transmitted bit of the destination field is the high order term of the message polynomial to be divided by G(x) producing remainder R(x). The high-order term of R(x) is the first transmitted bit of the Packet Check Sequence field. The algorithm uses a linear feedback register which is initially preset to all 1's after the last data bit is transmitted, the contents of this register (the remainder) are inverted and transmitted as the CRC field. After receiving a good packet, the receiver's shift register contains, 11000111000001001101101 01111011 (x^{31},.....,x^{0}).

Minimum Packet Spacing: This spacing is 9,6 us, the minimum time that must elapse after one transmission before another transmission may begin.

Round trip delay: The maximum end-to-end, round trip delay for a bit is 51,2 us.

Collision Filtering: Any received bit sequence smaller than the minimum valid packet (with minimum data field) is discarded as a collision fragment.

A1.? Control Procedure

The control procedure defines how and when a station may *transmit packets into the common* cable/fibre optic link (communication medium). The key purpose is fair resolution of occassional contention among transmitting stations.

Defer: A station must not transmit into the communication medium when carrier is present or within the minimum packet spacing time after carrier has ended.

Transmit: A station may transmit if it is not diferring. It may continue to transmit until either the end of the packet is reached or a collision is detected.

Abort: If a collision is detected, transmission of the packet must terminate, and a jam (4-6 byte of arbitrary data) is transmitted to ensure that all other participants in the collision also recognize its occurence.

Retransmit: After a station has detected a collision and aborted, it must wait for a random retransmission delay, defer as usual, and then attempt to retransmit the packet. The random time interval is computed using the backoff algorithm (below).

After 16 transmission attempts, a higher level (e.g. software) decision is made to determine whether to continue or abandon the effort.

Back-off: Retransmission delays are computed using the Truncated Binary Exponential Back-off algorithm, with the aim of fairly resolving contention among up to 1024 stations. The delay (the number of time units) before the 12th attempt is a uniformly distributed random number from (0 to 2n-1) for $0 < n < 10$ (n=0 is the original attempt). For attempts 11-15, the interval is truncated and remains at (0-1023). The unit of time for the retransmission delay is 512 bit times (51,2 s).

A1.3 Channel Encoding

Any type of encoding may be used, e.g. Manchester coding. Manchester coding has a 50% duty cycle, and insures a transition in the middle of every bit cell (data transition). The first half of the bit cell contains the complement of the bit value, and the second half contains the true value of the bit.

Data Rate: Data rate is 10M bits/s = 100 ns bit cell ±0,01%.

Carrier: The presence of data transitions indicates that carrier is present. If a transition of deferring is not seen between 0,75 and 1,25 bit times since the centre of the last bit cell, then carrier has been lost, indicating the end of a packet. For purposes of deferring, carrier means any activity on the communication medium, independent of being properly formed. Specifically, it is any activity on either receive or collision detect signals in the last 160 ns being properly formed.

Using a Coax C.

If a coax cable is to be used, the following configuration and specification are proposed.

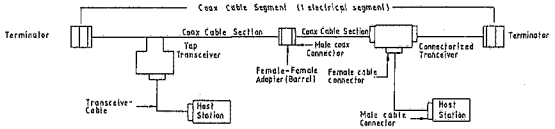


Fig. A1.2 Configuration - with coax cable

Coax Cable

Impedance : 50 ohms ±2 ohms. This impedance variation includes batch-to-batch variations. Periodic variations in impedance of up to 3Ω ohms are permitted along a single piece of cable.

Cable loss : The maximum loss from one end of a cable segment to the other end is 8,5 db at 10MHz (equivalent to 500 metres of low loss cable).

Shielding : The physical channel hardware must operate in an ambient field of 2 volts per metre from 10KHz to 30MHz and 5V/metre from 30MHz to 1GHz. The shield has a transfer impedance of less than 1 milliΩ per metre over the frequency range of 0,1MHz to 20MHz (exact value is a function of frequency).

Ground Connections : The coax cable shield shall not be connected to any building or AC ground along its length. If for safety reasons a ground connection of the shield is necessary, it must be in only one place.

Physical Dimensions : This specifies the dimensions of a cable which can be used with the standard tap. Other cables may also be used, if they are not to be used with a tap-type transceiver (such as used with connectorized transceivers, or as a section between sections to which standard taps are connected.).

Center Conductor:	0.0855" diameter solid tinned copper
Core Material:	Foam polyethylene or foam teflon FEP
Core O.D.:	0.242" minimum
Shield:	0.326" maximum shield O.D. (>90% coverage for outer braid shield)
Jacket:	PVC or teflon FEP
Jacket O.D.:	0.405"

Coax Connectors and Terminators

Coax cables must be terminated with male N-series connectors and cable sections will be joined with female-female adapters. Connector shells shall be insulated such that the coax shield is protected from contact to building grounds. A sleeve or boot is acceptable. Cable segments should be terminated with a female N-series connector (can be made up of a barrel connector and a male terminator) having an impedance of 50 ohms ±1% and able to dissipate 1 watt. The outside surface of the terminator should also be insulated.

Transceiver

Connection Rules: Up to 100 transceivers may be placed on a cable segment no closer together than 2.5 meters. Following this placement rule reduces to a very low (but not zero) probability the chance that objectionable standing waves will result.

Coax Cable Interface

Input Impedance: The resistive component of the impedance must be greater than 50 Kohms. The total capacitance must be less than 4 picofarads.

Nominal Transmit Level: The important parameter is average DC level with 50% duty cycle waveform input. It must be 1.025 V (41 mA) nominal with a range of 0.9 V to 1.2V (36 to 48 mA). The peak-to-peak AC waveform must be centered on the average DC level and its value can range from 1.4 V P.P to twice the average DC level. The voltage must never go positive on the coax. The quiescent state of the coax is logic high (0 V). Voltage measurements are made on the coax near the transceiver with the shield as reference. Positive current is current flowing out of the center conductor of the coax.

Rise and Fall Time: 25 ns ± 5 ns with a maximum of 1 ns difference between rise time and fall time in a given unit. The intent is that dV/dt should not significantly exceed that present in a 10 MHz sine wave of same peak-to-peak amplitude.

Signal Symmetry: Asymmetry on output should not exceed 2 ns for a 50-50 square wave input to either transmit or receive section of transceiver.

Transceiver Cable Interface

Signal Pairs: Both transceiver and station shall drive and present at the receiving end a 78 ohm balanced load. The differential signal voltage shall be 0.7 volts nominal peak with a common mode voltage between 0 and +5 volts using power return as reference. (This amounts to shifted ECL levels operating between Gnd and +5 volts; a 10116 with suitable pulldown resistor may be used). The quiescent state of a line corresponds to logic high which occurs when the + line is more positive than the - line of a pair.

Collision Signal: The active state of this line is a 10 MHz waveform and its quiescent state is logic high. It is active if the transceiver is transmitting and another transmission is detected or if two or more other stations are transmitting, independent of the state of the local transmit signal.

Power: +11.4 volts to +16 volts DC at controller. Maximum current available to transceiver is 0.5 ampere. Actual voltage at transceiver is determined by the interface cable resistance (max 4 ohms loop resistance) and current drain.

ISOLATION:

The impedance between the coax connection and the transceiver cable connection must exceed 250 Kohms at 60 Hz and withstand 250V RMS at 60 Hz.

Transceiver Cable and Connectors

Maximum signal loss = 3 db @ 10 MHz. (equivalent to =50 meters of either 20 or 22 AWG twisted pair).

Transceiver Cable Connector Pin Assignment

1. Shield*	9. Collisions
2. Collision +	10. Transmit
3. Transmit +	11. Reserved
4. Reserved	12. Receive
5. Receive +	13. + Power
6. Power Return	14. Reserved
7. Reserved	15. Reserved
8. Reserved	

*Shield must be terminated to connector shell.

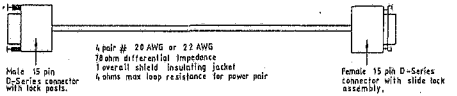


Fig. A1.3 Connectors with and without lock posts

APPENDIX 2

A DISTRIBUTED COMPUTER CONTROL SYSTEM (DCCS) ARCHITECTURE:

A2.1 The System Model:

Consider a process control plant as containing two sub-systems:

a) the physical plant system and

b) the control system.

A2.1.1 The Physical plant system:

This system comprises the physical plant facilities (construction, machinery, etc.) which are used in order to produce the desired end products from the input resources (raw materials, energy) which are available. This physical plant can be viewed from two different aspects:

a) the material transformation aspect and

b) the information transformation aspect.

From the point of view of material transformation the flow processing of materials and energy to generate the final product is considered.

From the point of view of the information transformation the processing of the information by the physical plant is considered.

A2.1.2 The control system:

This term is used according to its conventional meaning.

It comprises all information processing facilities which are used in order to control the physical plant system.

A2.2 *System Building Blocks*

The convenience of using building blocks in specifying centralized computer systems is well recognized, e.g. processing power, storage of programs and data, interfaces and peripheral units selection. The PMS (processor, memory, switch) classification is useful in defining the main building blocks but it must be a set of building blocks already accepted in centralized systems and networks is also applicable to the design of distributed systems. Fully recognizing that the sufficiency condition has not been met, a set of components is extracted, keeping in mind the conflicting requirements of simplicity and completeness. For the purpose of the present study, the following building blocks are adequate: processing unit, memory unit, topology, routing policy, line control procedure and data base structure.

A2.2.1 *Processing unit*

The processing unit (PU) is a component that is capable of performing a sequence of operations prescribed by a program. It includes limited information storage capacity (RAM).

A2.2.2 *Memory unit*

The main function of a memory unit is to hold a substantial amount of information unchanged over a period of time. The ability to recollect information at will is a necessary addition to this main purpose.

A2.2.3 Topology

Topology pertains to the manner in which the various components of a network of system are physically interconnected, with emphasis on the data control flow paths. It will be appreciated that a routing policy is needed to govern the selection of proper routes for the orderly flow of information. For instance deterministic policy is time invariant; it ignores load fluctuations and can achieve optimal routing under steady state conditions.

A2.2.4 Line control procedure

A suitable line control procedure is required to reduce errors that are introduced during data transmission. After the proper routes have been selected, the correct sequencing and the integrity of data transmitted between parts of the system are assured by the judicious application of controls (10,13).

A2.2.5 Data base structure

A data base structure encompasses the manner in which information is acquired, stored, and dispersed. It is intimately linked to the overall system architecture and will be identified later as a key to the design of distributed systems.

A2.3 Generalized Design Sequence

The generalized design sequence involves three major design areas namely:

a) Processing clusters
b) Data communication
c) Data base

A2.3.1 Processing clusters

A key activity in planning a centralized configuration is associated with the determination of the processing power required and the selection of a suitable processor. The ease of generating supervisory and application programs depends upon the availability of software tools and, to a certain extent, can be improved as new features are developed. Never the less, the ultimate performance and the expansion capability of the cluster is determined at the outset by the characteristics of the selected computer. The choice is complicated by the necessity to anticipate future needs and to assess carefully the status of technological developments likely to influence such parameters as memory address space and cycle speeds, word lengths, data types, versatility of instruction set, etc.

In this respect, the selection of processing units for distributed systems is not as crucial since changing requirements can be met by adding clusters in a manner compatible with the architecture. Each cluster is defined at the appropriate time with a high degree of confidence, using known methods already developed for centralized systems.

A2.3.1.1 Data communication

The term "data communication" is usually associated with networks consisting of loosely coupled geographically dispersed components. In contrast, coupling between parts in a distributed system is more specific and greater degree of interdependence prevails amongst components. The distinction between networks and distributed systems arises from the concern for communication problems in designing networks, due to the large seperation between parts.

The rationale behind the development of data communications networks is a combination of two factors: resource sharing and the ability to access common information remotely. The topology and the line control procedures are designed to minimize the communication costs, bearing in mind reliability and speed requirements. Since most computer networks rely upon the common carriers for information transfer, the method is heavily oriented towards tariff rate analyses, and error detection and correction techniques for telephone quality lines (9, 8, 19).

Clearly, the emphasis on common carrier line cost and quality is not pertinent to the design of distributed systems for process control. However, the understanding achieved in data networks of queing disciplines, traffic analysis techniques, protocols, etc. are directly relevant. In other words, new tools for distributed systems can be built based on a large body of information that is already available.

A2.3.2 Data base

In studying the first two areas of the design sequence for distributed systems, i.e. processing clusters and data communications, it was concluded that much experience can be derived from centralized systems and networks.

However, the design of a distributed data base is more complicated as it is influenced by the selection and specific implementation of a partitioning strategy. The magnitude of the difficulty should not be taken as an intrinsic property of distributed systems. To a large measure it reflects the currently limited understanding of distributed data bases compared to processing clusters and data communications.

For example, the engineering trade-offs between "physical partitioning" "and functional partitioning" have to be resolved both at the system and data base levels.

Physical partitioning of a data base recognizes the actual distribution of the process to be controlled and attempts to bring the storage closer to the areas of highest activity, i.e. near data sources and/or demands. On the other hand, functional partitioning tries to minimize cost of system hardware and simplify certain operations by grouping devices according to capabilities and functions. When carried to an extreme, this latter strategy implies the concentration of most data handling functions as in the centralized approach. To help identify the salient design issues, a representation of a generalized data base is built up in a series of four steps (figs. A2.1 to A2.4) starting with simpler, more traditional structures. The approach is similar to that followed in introducing the generalized design sequence.

The centralized data base structure, illustrated in fig. A2.1 lends itself naturally to a single processor architecture. The term "Data" collectively refers to all information, the associated storage devices scheme. The "Common access port" represents a hardware/software interface that governs the orderly bidirectional flow of requests and data from/to "Users" and the unidirectional flow from "Data Source" to "Data". For ease of description, the term user includes any person, program or piece of equipment needing access to some data. This data base representation is applicable to a large majority of present day systems.

When indirect (or remote) access to common data is necessary, the data base configuration remains essentially the same except for an additional hardware/software "Communications Interface" (a point illustrated in the correspondence between fig. A2.1 and A2.2).

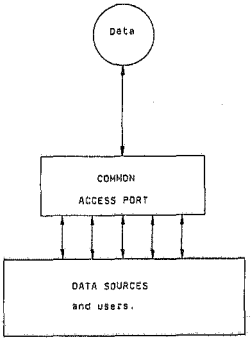


Fig. A2.1

A centralized data base.

The main function of the interface is to satisfy the line control procedures needed due to the remoteness of the Data Sources and Users.

The same representation can be simply duplicated (fig. A2.3) to illustrate resource sharing between seperate data bases. In order to permit uniform access of all data by all users, "Interpreters" must be introduced to translate the rules peculiar to each case, thereby achieving data base transparency. This is typically the situation prevailing in computer networks where the loose coupling between components permits the coexistence of independent data bases with their own Common Access Ports ("A" and "B" of fig. A2.3).

The form of fig. A2.3 suggests a possible representation of a generalized distributed data base (fig. A2.4). The higher level of commonality between "Local Data Bases" in distributed systems is accomplished by the introduction of "Common Ports" shared by all Sources and Users. The total capability of Common Ports must satisfy all information transfer requirements from/to and between Local Data Bases in a distributed system. The functions to be performed include, for instance, buffering and dispatching of requests, provision of communications medium and definition of uniform protocol. Hierarchically, the common ports are one level above the local data bases. In this manner, the prime problem in designing a distributed data base is reduced to defining common ports to provide access to a number of Local Data Bases. The problem is manageable as each data base preserves its identity and can in fact use a standard centralized design partly modified and augmented with the necessary interpreter. Difficulties could arise if total transparency between Local Data Bases were to be guaranteed in a truly dynamic real-time application. However, this requirement can be avoided by choosing a suitable partitioning strategy and by adding more clusters.

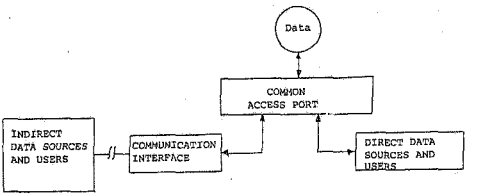


Fig. A2.2

Remote access to a common Data Base

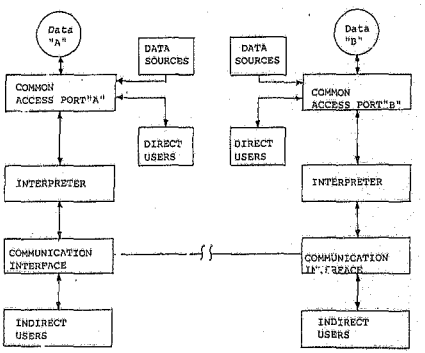


Fig. A2.3

Resource sharing between seperate data bases.

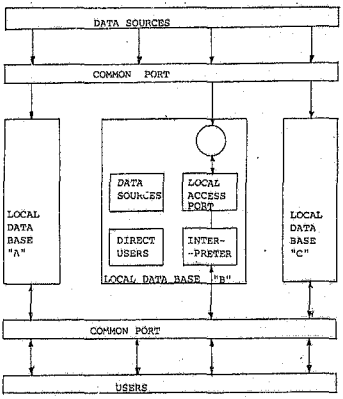


Fig. A2.4

A generalized distributed Data Base.

A2.4 Control Hierarchies

In addition to the possibilities mentioned in earlier sections for eliminating the disadvantages of the conventional methods for process control and the fulfilment of the higher requirements of processes, modern distributed process control also offers the particularly important possibility of constructing almost any complex control system in a manner that is clearly organised according to hierarchy stages. In order to do this, the process to be controlled, e.g. a power station , a chemical process, a process in the iron and steel or cement industries, or a traffic-control system is split into partial processes (see Fig. A2.5).

The partial processes are allocated to autonomous control units - so called - "autonomous islands" - a,b,....n of the process-oriented hierarchy level in which all control functions of associated partial processes are combined, e.g. open and closed loop control, monitoring, computing, etc. The expression "process oriented" is a functional term and is not to be understood as referring to the spatial arrangement of the control devices.

Automation islands in which the master and coordination functions of a large process are realized (e.g. A for the partial processes with the control units a,b, and c) form the coordination level (see fig. A2.5). The hierarchy peak (of a three stage system in this example) is the supervisory control level with superior control functions which affect the process.

A hierarchy structure has the following important advantages:

a) If only low demands are made on the control system , the omission of the higher hierarchy level presents a money-saving solution.

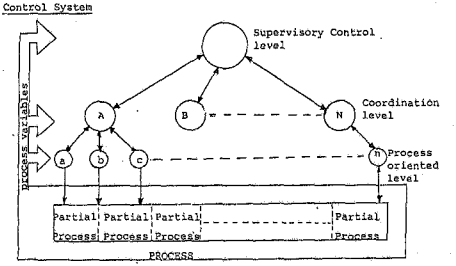


Fig. A2.5

Basic structure of a hierarchy control system divided according to partial *processes*.

b) Higher structural reliability of the control system can be achieved with autonomous automation islands allocated to the partial processes. The failure of one island only affects the partial process associated with it (switch to manual operation!). The other partial processes continue unaffected in automatic mode. Should a higher hierarchy level fail, the automatic operation of the total process is nevertheless retained with the help of the subordinate levels, although, of course with a lower degree of automation.

BIBLIOGRAPHY

1. Amendt A.J., "Microprocessors: The revolution in control system design", control and instrumentation, February 1976.

2. Bell C.G., Mudge J.C. and Mc Namara J.E., computer engineering, digital press, Bedford: Mass 1978, p.32.

3. Benke H., "A review of European safety code requirements for computerized instrumentation in Hazardous area chemical plants", Paper No. 99-111, 2nd Joint Spring ISA conference, April 1975.

4. Berret J.R., Hurd E.T. and Williamson W.E., "The special promise of microcomputers as Elements in distributed computer control systems", proceedings of the 3rd purdue university conference for advanced industrial control, April 1976.

5. Cartright L.M. and Mayberry J.C., "Distributed Systems", a new approach to process control, Paper No. 511, ISA/75 Annual conference, October 1975.

6. Chemical Week, "Getting computers out into plants", December 3 1975.

7. Colloquys 1975 "The process control Loop Five Years and Beyond", held by Fisher controls company, August 1975.

8. "Computer communication network", edited by N. Abramson and Kuo F.F., Prentice Hall, 1973.

9. "Data networks:analysis and design", third data communications symposium, St Petersburg, Florida, IEEE Cat. No. 73CH0820-4C, Nov. 1973.

10. Datapac: "standard network access protocol specifications", the computer communications Group of the Trans-Canada telephone system, 1976.

11. De Roze B.C., Nyman T.H., "The software life cycle", a management and technological challenge within the department of defence, IEEE TRANS. SE-4, July 1978, p.309-318.

12. Desktop computing "LOCAL AREA NETWORKS", p.30-33.

13. "IBM Synchronous data Link control general information", GA 27-3093-0, File no. GENL-09, March, 1974.

14. Inose F., Takasugi and Hiroshima M., "a digital data highway system for process control" - Paper no. 70-510, 25th Annual ISA conference; October 1970.

15. Kopetz H. "distributed computer control systems", 1981.

16. Kopetz H., Lohnert F., Merker W., "An outline of project MARS", Maintainable real time system, FB 20, Bericht der TU-Berlin, 1979.

17. Le Lann Gerold, "Distributed systems, towards a formal approach", 1977 IFIP congress proc. pp. 155-160.

18. Montgomery Philster Jr., "Technology and Economics: integrated circuit geometry and optimum chip size", Computer design, Nov. 1979, pp. 30-38.

19. "Network structures in an evolving operational environment", Fourth Data communications symposium, Quebec City, IEEE Cat. No. 75CH1001-7 Data, October 1975.

20. Noyce R.N., "From relays to MPUs", computer, Dec. 1976, pp. 26-29

21. Phillip H., Enslow Jr., "What is a distributed data processing system?", Computer, Jan. 1978, p. 13-21.

22. Philster M., data processing technology and economics, Santa Monica Publishing Co. 1976, pp. 180-187, 491-493.

23. Randel B., Lee P.A., Treleaven P.C., "reliability issues in computing system design".

24. Schoch J.H., Dalal Y.K., Redell D.D., "The evolution of the Ethernet local computer network", Xerox R.C. Crane, 3 Comm, IEEE 1982.

25. Simon H., "Multiplex system Save Multibucks in refinery and chemical plants", Paper No. 70-564, 25 Annual ISA conference; October 1970.

26. Terao M., Paper No. 108, "Automation Control symposium of the Society of automatic Control", Japan, March 1959.

27. Williams T.J., "The coming Years - the Era of computing Control", Instrumentation Technology, January 1970.

28. Yoshü., Kuroda S., "A distributed control system concept and architecture, proceedings of the IFAC workshop on distributed computer control systems", tempe, Fla, USA: 1979.