MODELING, IDENTIFICATION AND CONTROL, 1987, VOL. 8, NO. 4, 201–222
doi:10.4173/mic.1987.4.2

A truly distributed process control system with on-line configuration and expansion capabilities

STEINAR SAELID[†]

Keywords: *Process control system, distributed control*

The process control system described in this paper, is a very flexible system equipped with a range of computer-aided configuration tools. These are tools for the configuration and reconfiguration of control loops, computing blocks, logic control modules, i/o connections or graphics, and presentation elements such as reports, trend curves, flowsheets, free graphical pictures etc., while the system is operational and in control. Documentation of system topology, system parametrization and the i/o configuration/connection etc., can be extracted from the system at any time. The system can also be configured for training and simulation applications as well as for process control. The paper describes the system hardware architecture, the distributed database and the configuration and operation tools.

1. Introduction

Some desired qualities of a process control system are:

A self explaining, process oriented man/machine interface, preferably with help-functions to guide the user. The user of the system should be able to influence the configuration of the man/machine interface (MMIf). An operator should have the possibility to change and define his own 'views' into the process by defining his own pictures and his own graphical representations of the process.

Flexibility for easy change and expansion possibilities while the process and the process control system are in operation.

Good scalability characteristics. This means that a small increase in required capacity should be implemented with a small increase in hardware investment.

A uniform structure. This involves analogue control functions, ordinary monitoring functions, complex computational tasks, as well as logic and sequencing functions which should be implemented in the same process control system.

Good redundancy and backup solutions.

Self documenting capability. Complete or partial documentation of the control system should be obtained from the system on request. The documents should be suitable for support during installation, maintenance and change work, as well as for reference documentation of the total system installation.

The AIM system is a distributed process control system offering the above-mentioned qualities. The system is based on the most recent theories concerning both hardware and software. The result is a system with unique flexibility and operational capability:

Received February 1, 1987.

[†] Kongsberg Albatross a.s., Postboks 483, N-3601 Kongsberg, Norway.

Parameters as well as system structure and topology can be changed while the system is running. This is done by drawing of function blocks and connections between them on the graphical screen.

New i/o signals can be added or deleted while the system is on-line.

Logic control subsystems can easily be debugged and changed during normal operation. This is done by watching live diagrams on the screen. Digital signal values are shown by colour coding of signal lines.

The user can freely program his own digital, analog, or mixed digital/analog function modules by graphics programming. These functions can be stored in a library for later reuse. Programming, testing and installation of new function modules can be done while the system is controlling the process.

In addition to being a process control system, AIM can also function as a simulator. This provides comprehensive facilities that enable personnel to train on a model of the process before the actual process is operational, and also enables different forms of operator communication to be tried out in order to find out which is the most suitable. When the system is on-line, parts of the system can also be used as a simulator, with the operators using the same display screens, and input units that are used during normal operations. The only difference is that 'process values' are obtained from models of the process instead of the actual process.

2. System overview

AIM operates on two levels, the process level, and the supervisory level. One level comprises one or more stations of different types, depending on the operations required to be performed. A collection of such stations, which are logically related and connected by a data communication link, is termed a segment.

The stations are the main building blocks in the system (Fig. 1). These are of different types, each type designed to cope with a specialized set of tasks. A station consists of one or more processor cards, storing modules (disks), displays, input devices for operation, and process i/o cards. The mix of the components depends on what the station is intended to do.

Stations are located at different places in the plant, and they communicate with each other by a communication link. Stations on the same segment communicate directly. Stations located at different segments communicate via communication bridges and the supervisory network.

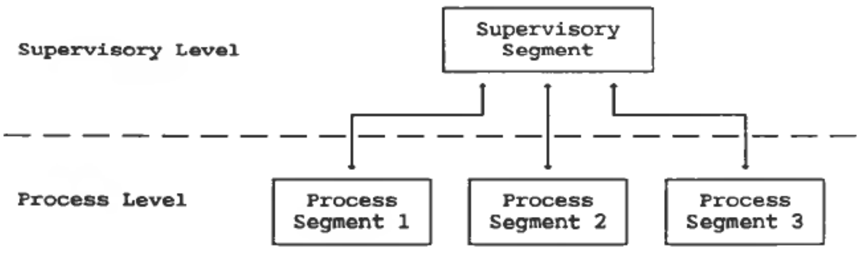


Figure 1. System structure.

2.1. The system levels

2.1.1. The process level

This is the basic level. Stations at this level may have direct access to the process through the process i/o. Such stations perform control actions, acquire and store process data and support communication with other stations. The operator is able to change control strategies or the graphics layout while the process is running.

If the process consists of a huge amount of i/o signals, or is geographically distributed, it may be convenient to divide the AIM-system in segments reflecting the process structure. This may also be advantageous for safety or operational reasons.

2.1.2. Supervisory level

This level is normally not connected to the process through the process i/o, and will often be omitted in small systems. All stations on this level are connected to the same segment, i.e. only one supervisory level segment is allowed. From this level the operators may access data from all other segments, and take control in process areas if desirable. This action may be denied, or controlled by a keylock or by passwords. Work on this level could typically be optimization, statistics and estimation tasks.

2.2. Main building blocks

All stations consist of one or more single board computers, named SBC1000. This computer is built around Intel's iAPX 186/87 processors, and may be connected to Winchester hard disks and/or floppy disks. The stations communicate through a 10 Mbit/s channel (Ethernet, IEEE 802.3).

The operator station (OS) (Fig. 3).

This is the operators 'window' into the process, and is equipped with up to two colour displays. It may be connected to printers and a colour hard copy unit. The operators communicate with the process through a tracker ball, function button panels, and user defined buttons. An alphanumeric keyboard is used for configuration purposes.
The operator station uses two SBC1000s, one to take care of the communication from the operator and to/from the disks, the other communicates with a display controller, printers and the tracker ball.

The process station (PS) (Fig. 4).

The PS performs the process related tasks, collects i/o, carries out signal conditioning, and local storage of data.

In addition to performing control and i/o-processing, the PS can handle PLC-functions which are normally done in dedicated stations in the comparative systems available today.
One PS may consist of up to 5 SBC1000s, depending on the number of i/o-signals connected and the computing power needed. The station may be connected to a local operator panel to handle simple local control, and act as a backup for the OS.

Remote i/o station (RIO)

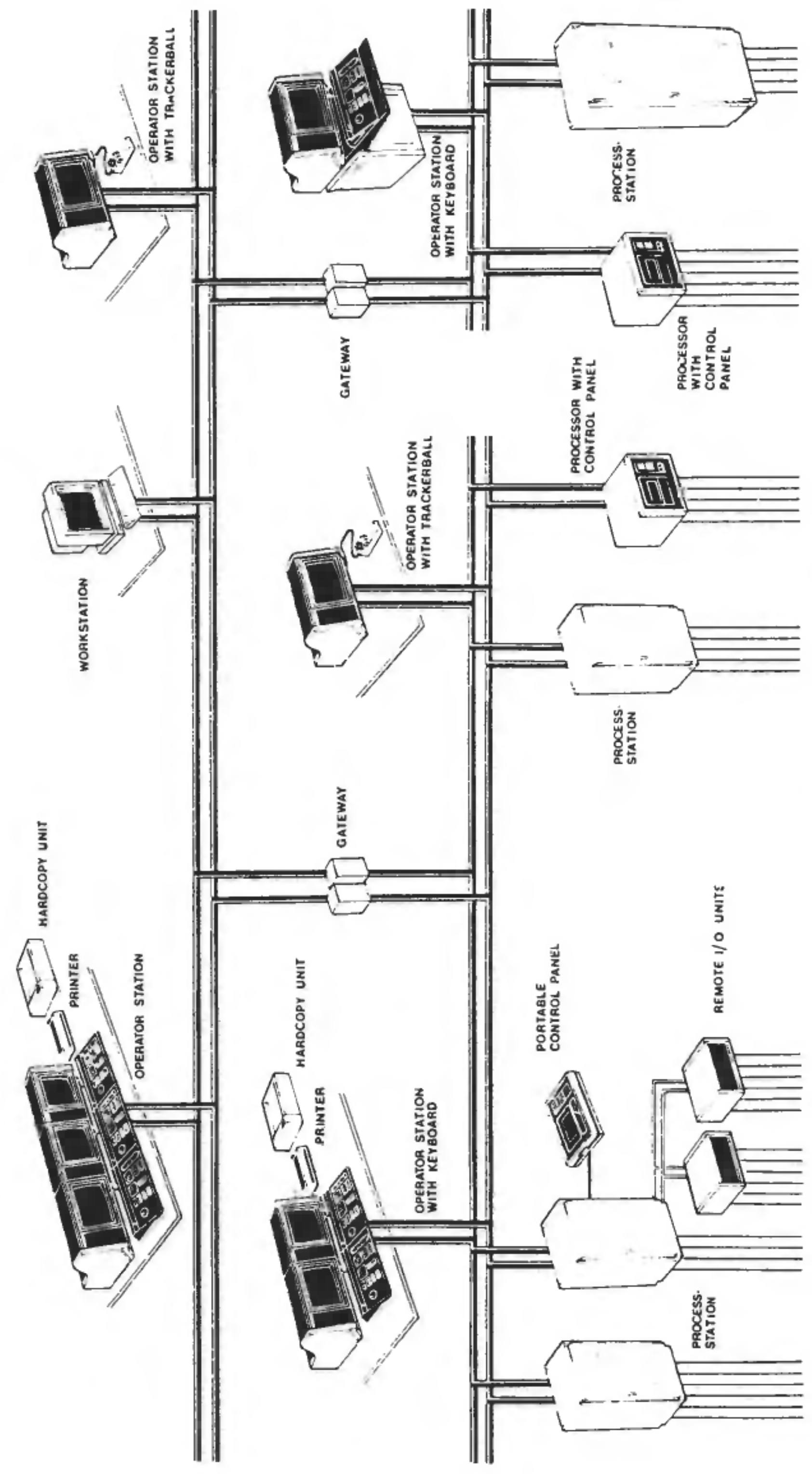

OPERATOR STATION WITH TRACKERBALL
OPERATOR STATION WITH KEYBOARD
PROCESS-STATION
PROCESSOR WITH CONTROL PANEL
GATEWAY
WORKSTATION
OPERATOR STATION WITH TRACKERBALL
PROCESSOR WITH CONTROL PANEL
PROCESS-STATION
GATEWAY
HARDCOPY UNIT
PRINTER
OPERATOR STATION
HARDCOPY UNIT
PRINTER
OPERATOR STATION WITH KEYBOARD
PORTABLE CONTROL PANEL
REMOTE I/O UNITS
PROCESS-STATION

Figure 2. General Layout.

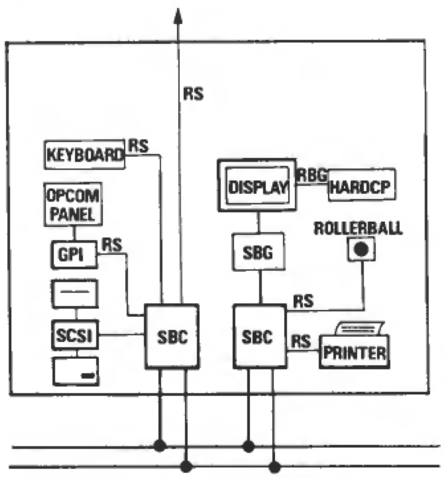


Figure 3. Operator station.

This unit is connected to a PS by a local serial bus (Bitbus) and is used to collect data which is distant from the PS. Several RIO-units may be connected to the same PS.

Communication bridge (BG)

This station connects segments on different levels.

2.3. Main design principles

One of the main goals when designing AIM has been to achieve flexibility with respect to both function as well as capacity.

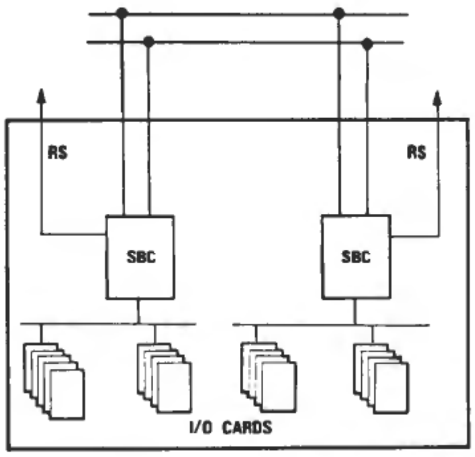


Figure 4. Process station.

2.3.1. The totally distributed database

The system has a completely distributed database. Configuration data, graphics data, process data and programs are located in the local process stations, not in a central database.

2.3.1.1. Scaleability. A consequence of the completely distributed database is a scaleable technology. In a scaleable system the price as a function of capacity is approximately linear over a considerable capacity range.

2.3.1.2. Process independent operator station. In most systems presently available, the operator station stores a lot of process dependent data elements, such as alarm lists and graphics related information. This represents a nonscaleable technology, because the operator station will reach its capacity limit sooner or later.

In the AIM system, all types of data are totally distributed to where they are generated and naturally belong. As an example, a trend curve generated by a level transmitter is stored in the PS where the module handling the level signal is located. This makes the OS a general device which does not contain any process specific information. If a picture is asked for, this is requested via the communication network, and the appropriate data is returned to, and interpreted by the OS software. Because every OS in a system can be made identical, each OS can take over the function of any other OS.

2.3.1.3. Single storage of data elements. A problem in many systems is data duplication and data inconsistencies. This arises when data is stored both in the process station and in one or more of the operator stations. If a parameter is changed in one part of the system and not in the other, the same data element may have different representations in the same system. This is avoided in the AIM system because process related data is only stored at one place. (Except for redundancy and backup purposes).

2.3.2. The module concept

The control and simulation functions in a process station are built up from basic functional units called MODULES. In each PS there is a library of such modules (Fig. 5), which a user can utilize to build his system. Each module consists of a graphics symbol, an algorithm and an associated data structure which is special to that module. Some examples of modules are: a PID controller, a Kalman filter, a module for handling of valve operation, a level measurement module, a PLC-logic module or a sequence control module.

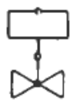
Each module has a number of TERMINALS. Terminals can be regarded as ports for communication between modules, and between a module and the i/o system. When a system is running, data will flow via the output terminals and input terminals of modules, and between the process terminals and the i/o system.

2.3.3. Flow sheet configuration

A key principle has been to design the system so that the configuration and operation of the system can be done by non-programmers.

Because a natural communication medium between a process engineer and a control system designer is the process flowsheet and the P&I drawings, a special mix of a process flowsheet and a P&Id is chosen as the basic configuration tool in the

THE MODULE CONCEPT

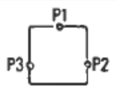
SYMBOL WITH TERMINALS	ALGORITHM FOR	DATA
	Control computations Signal conditioning Simulation	Parameters Graphics Info Alarm Info Variables

TERMINALS

In-terminals:	Connection point for computed signal. The signal value is fetched from an out-terminal on another module. Examples are: • Setpoint terminal of a controller. • Open/close command terminal of a valve.
Out-terminals:	Connecntion point for computed signal. The signal value at an out-terminal is generated by the module itself. Examples are: • Output signal from a controller. • Feedback signal from a valve. • Output from a level measurement module.
Process terminals:	Connection point between the module and the I/O system. Also used for connection between a control/instrument module and a simulator and between simulator modules.

EXAMPLE

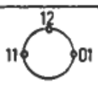
TANK MODULE



P1 — Level measurement process terminal

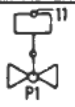
P2, P3 — In/Out flow process terminals

PID CONTROL MODULE



11 — Setpoint input terminal
12 — Measured signal input terminal
01 — Controller output signal

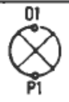
CONTROL VALVE MODULE



11 — Command input terminal

P1 — Flow process terminal

LEVEL MEASUREMENT MODULE



P1 — Process terminal for process or sinulator input

01 — Calculated level

Figure 5. The module concept.

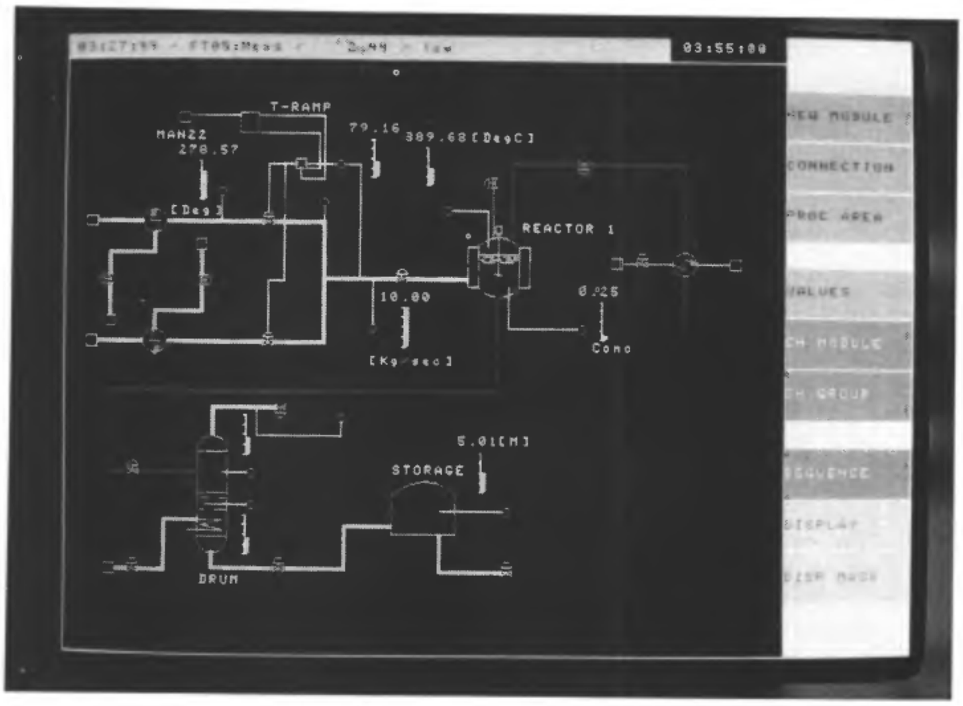


Figure 6. Flowsheet.

system. We shall call this special flowsheet the system flowsheet (Fig. 6). At configuration time, the user is simply putting the system together by drawing the system flowsheet on the graphics screen.

When an operator selects a module from the library and fits it into the system flowsheet, a copy of the module's data structure is automatically created in the selected PS. When the operator makes a graphic connection between the terminals of two modules, the system automatically sets up data references between the modules so that the system itself arranges for the proper data transfer between the modules when the system becomes active. This occurs even if the modules are located in different process stations. In such cases the system arranges for repetitive data transfer through the data communication network.

The total system is defined by a large virtual system flowsheet. The operator can view parts of this system flowsheet on the screen at any time, and address modules in the system flowsheet by the tracker ball controlled cursor.

2.3.4. *Softkey controlled MMIf*

A design goal has been to create a man/machine interface which is easy to adapt to a specific application. This is achieved by constructing a soft key based MMIf and a computer tool with which the user may create different MMIf systems.

2.3.5. *Software design principles*

The software implementation is based on advanced programming concepts such as object oriented programming and remote procedure calls. This produces a highly 'orthogonal' code with few and well-defined interactions between different system parts.

3. Description of the configuration tools

3.1. Man-machine interface editor (*BED*)

This editor is used to design and generate a tree-structured man–machine interface for the operator station. Each level of command contains several buttons. Each button may represent an action and a transaction to the next level. The action is a user written routine executed by the system when the button is 'pushed'. This action may be made system dependent, i.e. whether the routine is executed or not depends on which state the overall system is in. In addition, the next level could be system dependent as well.

Each button is described by attributes, such as colour, blinking status and access level. Different operator categories may be given access to different levels in the system. Buttons at higher access levels are not seen at lower levels, even if they occur at the same tree level.

A button is defined as either 'hard' or 'soft'. 'Soft' buttons are accessible only at the levels in which they occur on the display, 'hard' buttons are accessible in many or all levels. These are connected to dedicated buttons on the operator station, while 'soft' buttons appear on the right part of the display. The 'soft' buttons are selected through nine dedicated buttons on the operator station keyboard.

This implies that access to the alarm system for instance could be made by a 'hard' button which is active at every level, while the tuning of controllers has to be done by entering the appropriate level using 'soft' keys.

The editor generates C-source code, extracting information from the data structure generated by the user in the editor. This code is compiled and linked into the operator station software together with the rest of the system software.

3.2. Graphics editor (GED)

The graphics editor generates symbols for use in the system flowsheet. Examples of such symbols are valves, tanks, compressors, vessels and pumps (Fig. 7). These symbols may be generated from scratch, by drawing them on the screen, by editing existing ones, or by combining existing symbols with new ones.

The symbols are identified by name and number, and are stored in libraries for later use. Each new application will pick symbols from existing libraries, and create new symbols if they are not present.

3.3. Module editor (MED)

The module editor creates the skeleton of new data modules used in the process station. The user describes the module, in terms of number of process, output, and input terminals. These terminals are points where other modules in the flowsheet can be connected. A module has a name, and the module is connected to a symbol made by GED. In addition, the user connects an algorithm and a data structure to

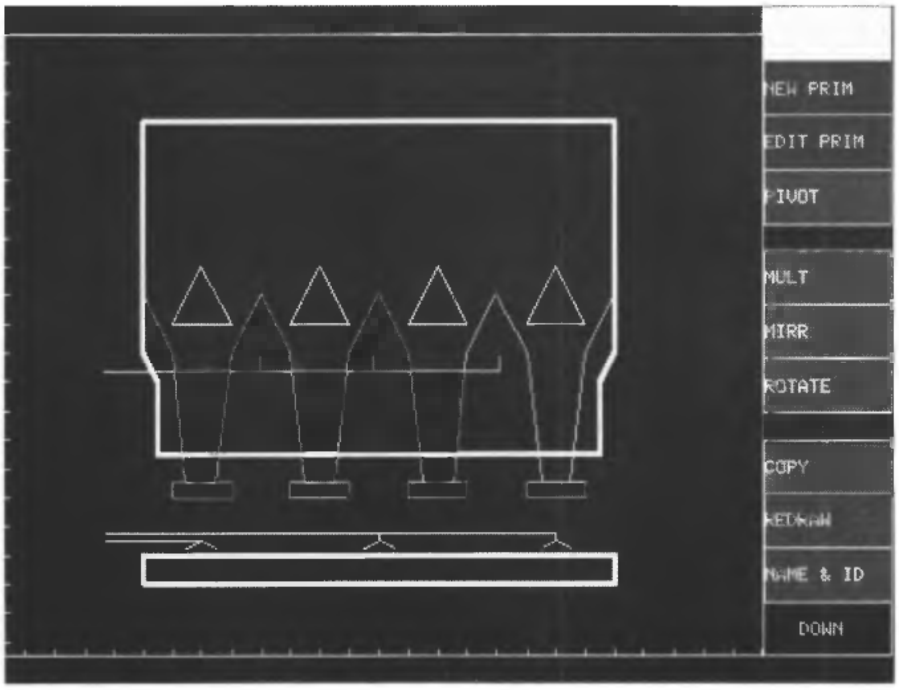


Figure. 7. Example. GED picture.

the module. This algorithm has a standard interface to the process station software through a code generated by MED. A module may, or may not have an associated algorithm. The latter is the case if the module is used only as a process equipment symbol. When simulation possibilities are wanted, the process equipment modules will have algorithms connected to tham as well.

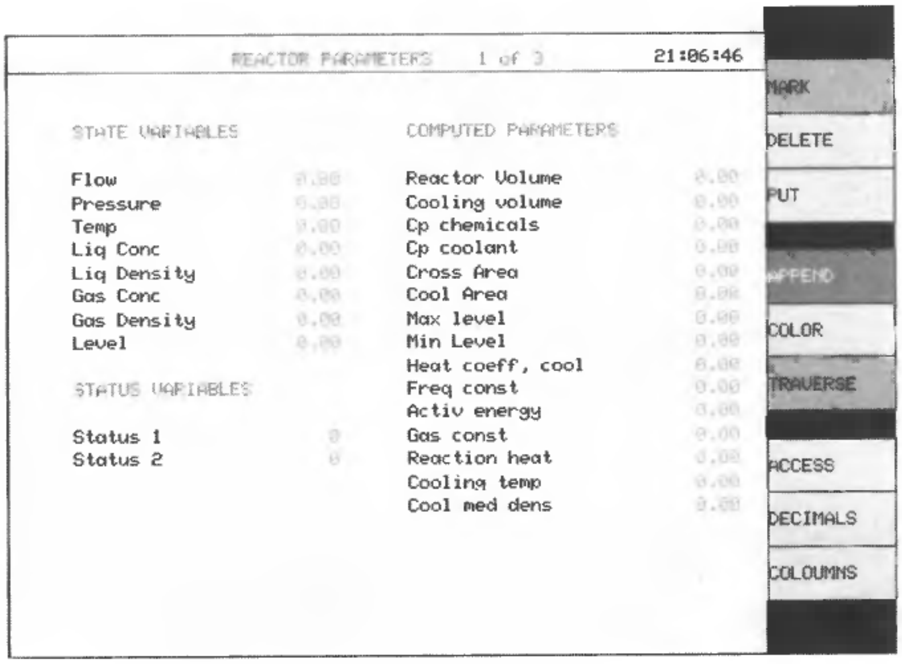
MED makes the basic representation of the module. When configuring the flow-sheet, copies will be made from this master-module (Fig. 8).

3.4. Report editor (RED)

RED generates the layout of standard reports and parameter pictures in the system. This is done by using information about the modules, and/or by changing already existing reports. The connection between reports made in this way, and the process, is done on-line, when the system is running or acts as a simulator. Reports can also be configured on-line.

3.5. Supplier configuration

This type of configuration may or may not be needed, depending on the degree of user requirements. In this case the tools described in the previous sections are used to generate the MMIf system, the special modules and the special reports required to implement the user's needs. At this level it is possible to try out different user interfaces, in close cooperation with the user.



```
              REACTOR PARAMETERS    1 of 3              21:06:46

   STATE VARIABLES                  COMPUTED PARAMETERS

   Flow             0.00            Reactor Volume          0.00
   Pressure         0.00            Cooling volume          0.00
   Temp             0.00            Cp chemicals            0.00
   Liq Conc         0.00            Cp coolant              0.00
   Liq Density      0.00            Cross Area              0.00
   Gas Conc         0.00            Cool Area               0.00
   Gas Density      0.00            Max level               0.00
   Level            0.00            Min Level               0.00
                                    Heat coeff, cool        0.00
   STATUS VARIABLES                 Freq const              0.00
                                    Activ energy            0.00
   Status 1            0            Gas const               0.00
   Status 2            0            Reaction heat           0.00
                                    Cooling temp            0.00
                                    Cool med dens           0.00
```

Figure 8. Example. MED picture.

3.6. Basic on-line configuration

Trends, group pictures and changes/additions in the system flowsheet etc. may be generated on-line while the system is running, or before start-up.

In principle, all operators could make their own pictures the way they wish to see the process. This will seldom be desirable in practice, due to the fact that other operators would be confused if the pictures were changing all the time. Therefore, parts of the configuration possibilities are normally restricted by keylocks or passwords.

3.6.1. I/O configuration

The configuration of the i/o system is also an interactive process whereby a particular module which is required linked to the i/o system is selected by pointing to the module in the system flowsheet. Thereafter the operator must similarly identify the process terminal that he wants to connect to the i/o system (if the module has several). The process station containing the module, will then search its own i/o system for cards of the correct type which also have free channels, and will present this information as a menu to the operator. The operator can then choose a card, and select one of its free channels.

Complete data for the selected card will then be displayed in tabular form, and the operator can fill in the required configuration data for the selected channel (name, type, scaling factors, terminal block connection, etc.).

The i/o system can be re-configured or expanded in a similar manner while the system is fully operational.

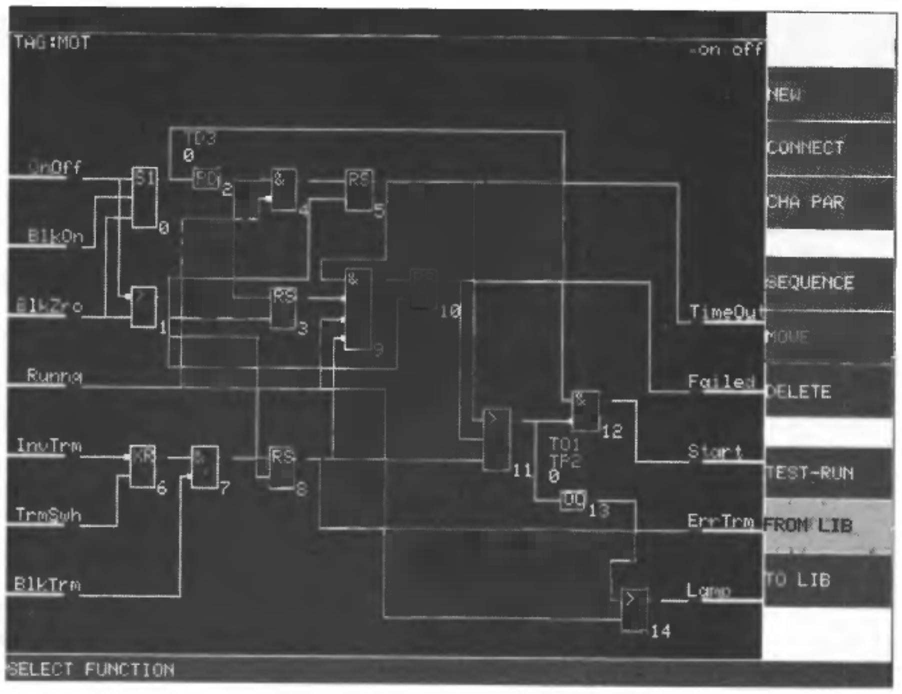


Figure 9. Example of a logic module diagram.

3.6.2. *Configurable modules*

Most system modules represent rather fixed types of functions, since a controller or a signal conditioning module has a fixed structure. Only the module parameters need to be changed or tuned.

However, some special modules are defined with a larger degree of flexibility. These modules can be quite freely programmed by the user by drawing elementary function blocks on the screen and connecting these together. By using these types of modules the user can configure his own special modules at any time. Typical applications are:

Logic control and interlock functions.

Sequence applications.

Mathematical computations.

Mass or energy balance calculations.

Mixed analog and digital control.

These modules have the same appearance in the system flowsheet as the fixed modules (Fig. 9). They have a symbol and a set of terminals. On configuration time, however, the user is asked to specify the number and types of terminals.

While the definition of a fixed module involves only parametrization and the connection of the module to other modules and/or the i/o system, the definition of a configurable module also means definition of the internal structure of the module. Each type of configurable module has its own range of elementary functions. These can be freely selected, within the capacity limit of the module, and connected

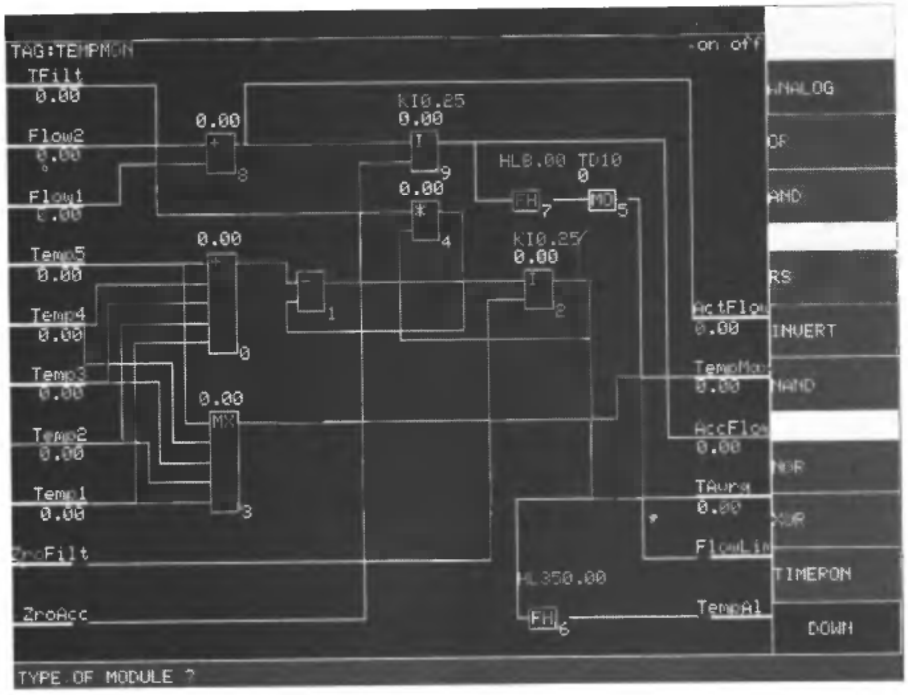


Figure 10. Example analogue digital module diagram.

together in a graphics picture in much the same way as the system flowsheet configuration.

This lower level module picture is termed a MODULE DIAGRAM. In other words, when a configurable module is defined, the user selects, positions and connects symbols representing the elementary functions in the module diagram. If the elements have parameters, these are given to them as they are configured into the module diagram. Each element can be connected to terminals of the other elements or to the module's input or output signals.

A configurable module can be stored in the operator station and duplicated for later use. A frequently used logic module, such as a pump control unit, can be constructed once and for all and reused every time a similar pump control module is required.

3.6.2.1. The logic elements. Examples of the basic logic elements are AND, OR, NAND, NOR, XOR, timers, counters, RS-elements, sequencers, etc. A typical logic module diagram is shown in Fig. 10.

3.6.2.2. The analogue elements. The main analogue elements are the usual arithmetic functions, a selection of mathematical functions, numeric integration, simple filters etc. as well as hybrid elements (comparator, rate detection etc.).

By using these elements, advanced calculations of dynamic and static type can be realized. Examples are mass and energy balance calculations, lead-lag compensation networks, Kalman filters, unit process simulators etc.

An example of a combined analogue digital module is shown in Fig. 11. The configurable modules allow an extreme flexibility. All modules can be re-configured or changed while the system is in operation, and the values can be observed in the

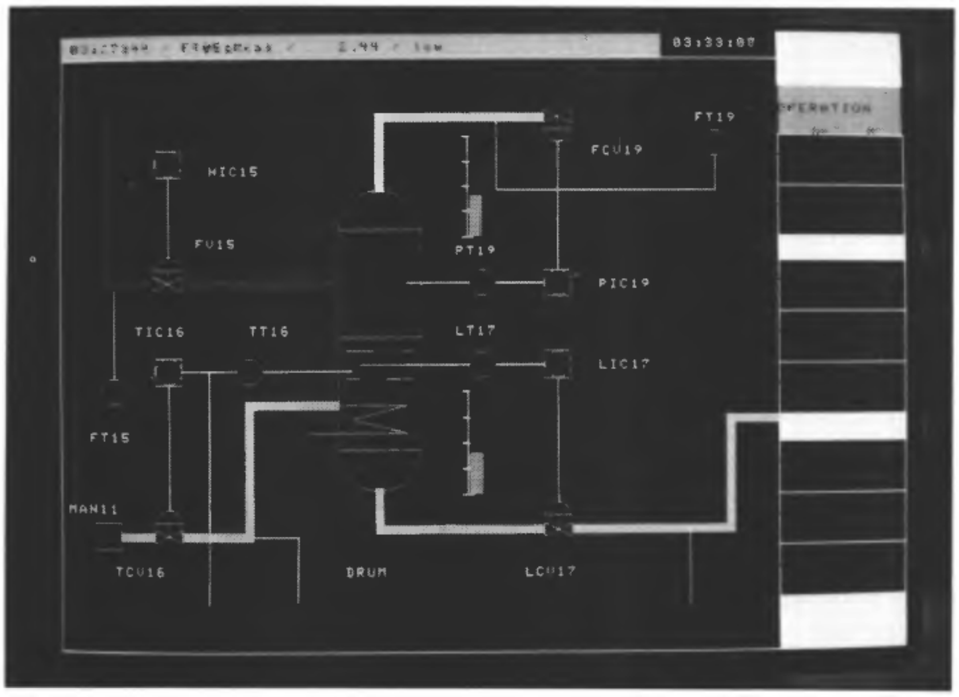


Figure 11. Example: Flowsheet detail.

module diagram while the system is running. As an example: In the logic module diagram, a connection is green if the signal is 'on' (1) and red if the signal is 'off' (0).

In and out signals in the module diagram are marked with names for easy recognition by the user.

4. Operational aspects

4.1. System downloading

Each system is delivered with the executable programs resident on the system's hard disks. At power up, the operator station computer will automatically download its own program from disk.

At power up on other stations, these will request uploading services on the network. As soon as an operator station with the appropriate fileserver and the requested programs is operative, the requesting station's program will start downloading automatically.

When a station's program is downloaded, the station is ready to be configured. In most cases configuration data resides on the hard disk, and configuration data is automatically downloaded to the requesting station.

4.2. Command changeover procedure

The flexibility and generality of the operator station makes it possible to access any subsystem from any operator station. There may also be several operator stations in a subsystem. In order to change control from one OS to another, the operator at the controlling OS usually has to release his control access. When this is done, any OS can take control responsibility by pushing a 'take-control' button.

In an emergency situation any operator station can get control responibility by force. This forced control transfer may be protected by a password or by key access. Any operator station has access to the monitoring and display functions at any time.

4.3. Response and sampling time

Each process station can run several tasks at different sampling speeds. Up to 10 different tasks all with a different sampling frequency may be running simultaneously. The sampling times may be selected from 10 ms to very large values. Every module in a process station may be assigned to run under any of the 10 application tasks. The different tasks can be assigned different priorities.

4.4 Capacity

Normally systems have limitations on the number of pictures and points they can handle. AIM differs from these systems as it gives no simple answer. All available space in a station will be used by a mix of continuous control functions, logic control functions, general computing tasks, pure monitoring tasks and for the storage of graphical information. That means that if all this space is filled with control information, there will be no room for graphical information. A typical mix in a PS could be

100 modules of typical continuous control

300 basic logic functions for logic and sequence control

30 trends

12 group pictures

1 overview picture

10 report pictures

In addition, the system flow sheet, the parameter pictures, the eventlog and alarmlists are stored in the process station. It is also important to notice that a module may handle more than one process signal, that is, one module is not equal to one process value.

Due to the modularity, any extension will only concern the process station involved.

4.5. Control/supervision mode

In this mode AIM collects data from the process i/o to control the process. The operator communicates with the process through the operator station using the tracker ball and the keyboard, or through local panels.

The flowsheet is the basic system picture, and can be scrolled and zoomed in and out to reach the part of the total picture that is of interest. It is possible to control the process from this picture, and values can be added and deleted in the picture on-line.

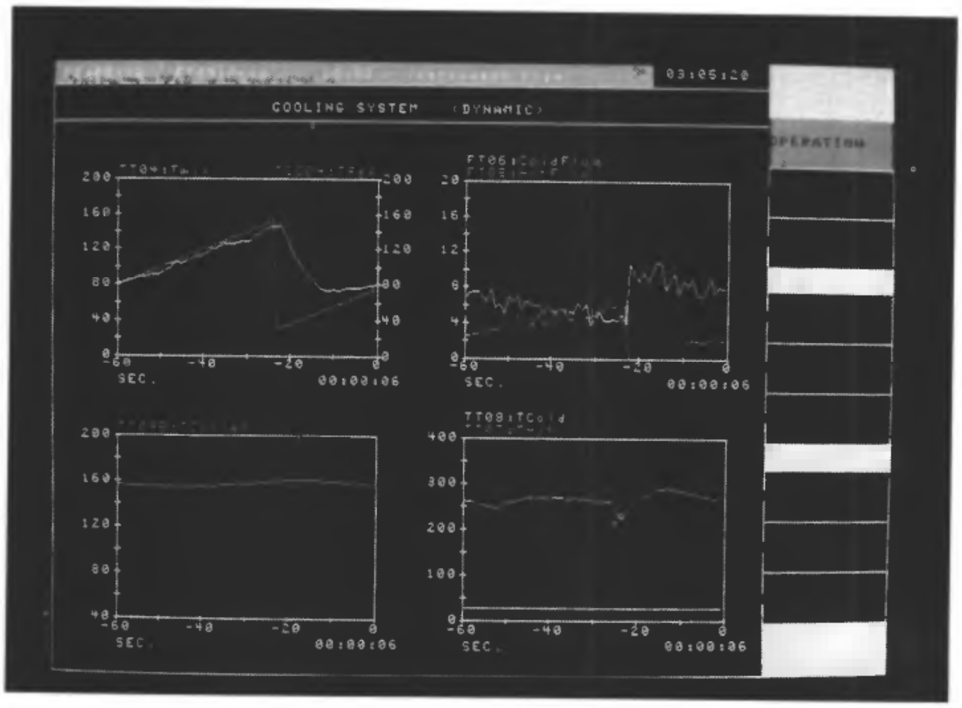


Figure 12. Example: Trend.

Other available pictures are trends, group pictures, parameter pictures etc. These pictures may be created on-line while the process is running, and the operator can access and control the process from them (Figs 12 and 13).

Alarm lists and event logs are always present in the system, and the operator has the ability to take printouts of these pictures. Reports and parameter pictures can also be printed.

4.5.1. Simulation mode

In this mode, all i/o-signals are computed by algorithms connected to each module. The system takes care of all administration concerning sequencing, synchronization etc. The operator can access the system in the same way as for the control mode.

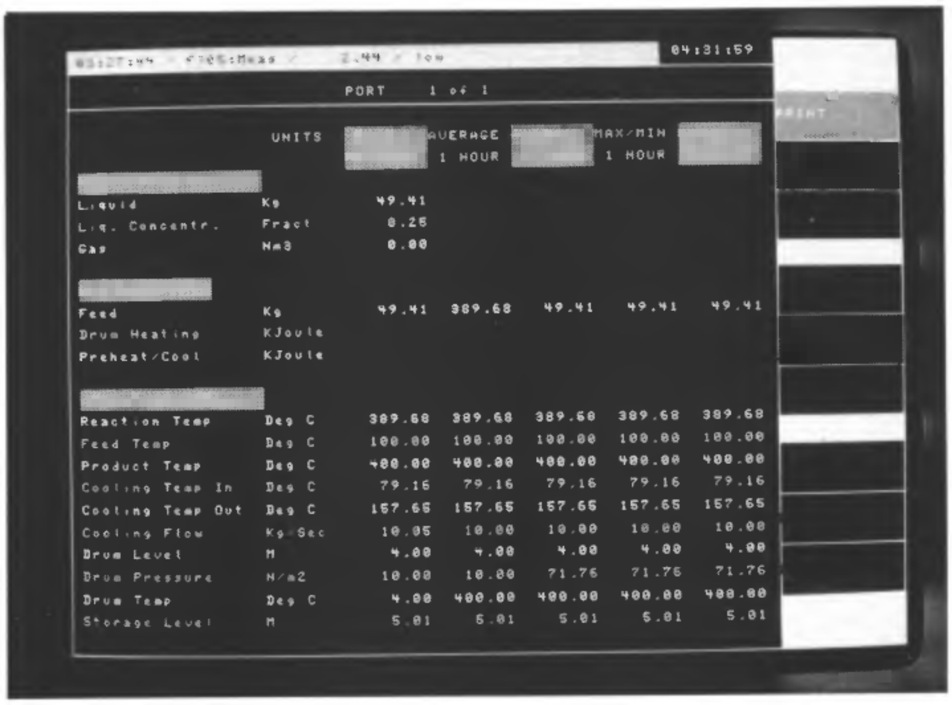
5. Increased efficiency and safety due to the use of CAD tools

5.1. Configuration

When the tools described above are used for designing and implementing an application, the efficiency of this task improves considerably. This is because of the modular design which makes it possible to build libraries for later use.

The use of the computer-aided-tools in the configuration phase significantly reduces the probability of introducing errors. These tools carry out extensive checking of the configured system, and tell the user what is wrong and how to correct it.

Since all hardware and software components can be tested against a simulator before the physical process is installed, the users are given the possibility to try



[illegible] 04:31:59

PORT 1 of 1

	UNITS		AVERAGE 1 HOUR		MAX/MIN 1 HOUR	
Liquid	Kg	49.41				
Liq. Concentr.	Fract	0.25				
Gas	Nm3	0.00				
Feed	Kg	49.41	389.68	49.41	49.41	49.41
Drum Heating	KJoule					
Preheat/Cool	KJoule					
Reaction Temp	Deg C	389.68	389.68	389.68	389.68	389.68
Feed Temp	Deg C	100.00	100.00	100.00	100.00	100.00
Product Temp	Deg C	400.00	400.00	400.00	400.00	400.00
Cooling Temp In	Deg C	79.16	79.16	79.16	79.16	79.16
Cooling Temp Out	Deg C	157.65	157.65	157.65	157.65	157.65
Cooling Flow	Kg/Sec	10.05	10.00	10.00	10.00	10.00
Drum Level	M	4.00	4.00	4.00	4.00	4.00
Drum Pressure	N/m2	10.00	10.00	71.76	71.76	71.76
Drum Temp	Deg C	4.00	400.00	400.00	400.00	400.00
Storage Level	M	5.01	5.01	5.01	5.01	5.01

Figure 13. Example: Report.

different control strategies and train their operators. This will lead to a more optimal system, and less problems in the start-up phase of the process plant. It also reduces the cost of having operators train on a process which is not completed.

If the system should be re-designed later on, most pictures and information could be corrected on-line. This will reduce downtime and re-configuring costs.

6. Redundancy and fault tolerance

The AIM system is designed to minimize the effect of faults and errors. If a fault occurs, the consequences should be as limited as possible, and actions must be taken to isolate the fault.

6.1. Fault isolation by distribution

Each single board computer is separately powered and electrically isolated from the rest of the system. The i/o cards are sectioned in groups of 11 cards, or optionally in groups of 5 cards. Each group has its own power supply, so a power supply failure will only affect cards within a group.

6.2. I/o card diagnostics

These types of checks are for optimal inclusion in a system, depending on the selection of i/o card types. The checks are based on the readback of output signals and on switching input channel relays for reading known test signals.

If an error is detected, this is reported as a system alarm. The alarm message contains a fault type indication, channel identification, card number and process station ID.

The card tests can be run with a frequency as high as 10 Hz. A typical test frequency in a real system would be in the range 0.1–1.0 Hz.

6.3. I/o power monitoring

The power supply for each group of i/o cards has a power-ok output signal. This signal is read back to the computer and used for system alarm generation if there is a power failure.

6.4. Dual system redundancy

The dual redundant system is designed to handle the following types of errors.

Computer failure detected by a watchdog system.

Computer failure detected by the accompanying processor.

i/o system power failure.

i/o card errors detected by self checks.

If an error occurs, the system will try to maintain its functionality by switching the master responsibility for a failed function to the slave computer (if this is working). A dual redundant system can be designed with complete or partial redundancy.

The system is designed so that externally, a dual computer pair looks like one computer. All external communication is always with the master computer.

6.4.1. Completely redundant system

In a completely dual redundant process station (PS), two identical computers with identical i/o systems are working in parallel. One part of the system is defined as the master system, the other as the slave system. A completely dual system is shown in Fig. 14. Input signals are branched to a dual set of input cards from the termination boards. The termination boards have termination blocks for the process signal connections, and a dual set of flat cable connectors to interface to the dual set of input cards.

Output signals or signals which are powered from the process station, require a switching method. The reason is of course that only one of the dual computers can control the process devices. The switching is done by relays and simple logic circuits mounted at the termination boards. Switching is done depending on the value of watchdog signals, power-ok signals and software-controlled signals from both systems in the dual configuration.

6.4.2. Partially redundant system

In a partially redundant system, a back-up computer is duplicating only some of the functions in one or more master computers. The i/o cards corresponding to the backupped functions must of course also be duplicated. A partially redundant system is shown in Fig. 15.

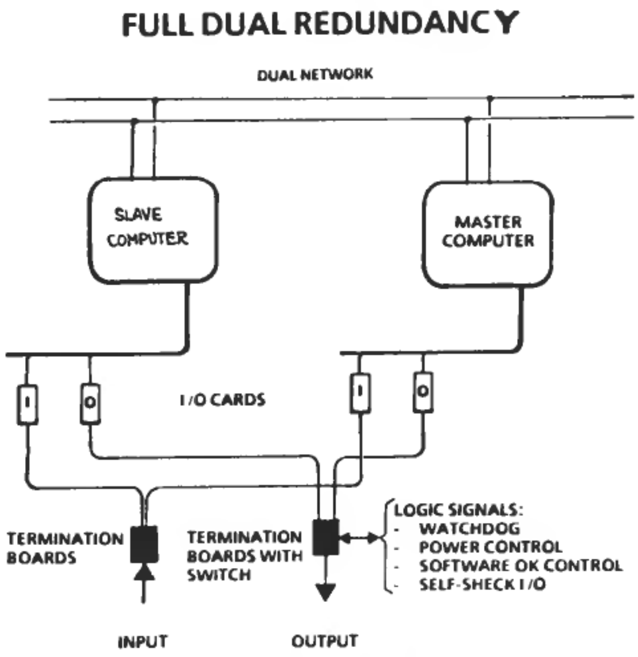


Figure 14. Completely redundant system.

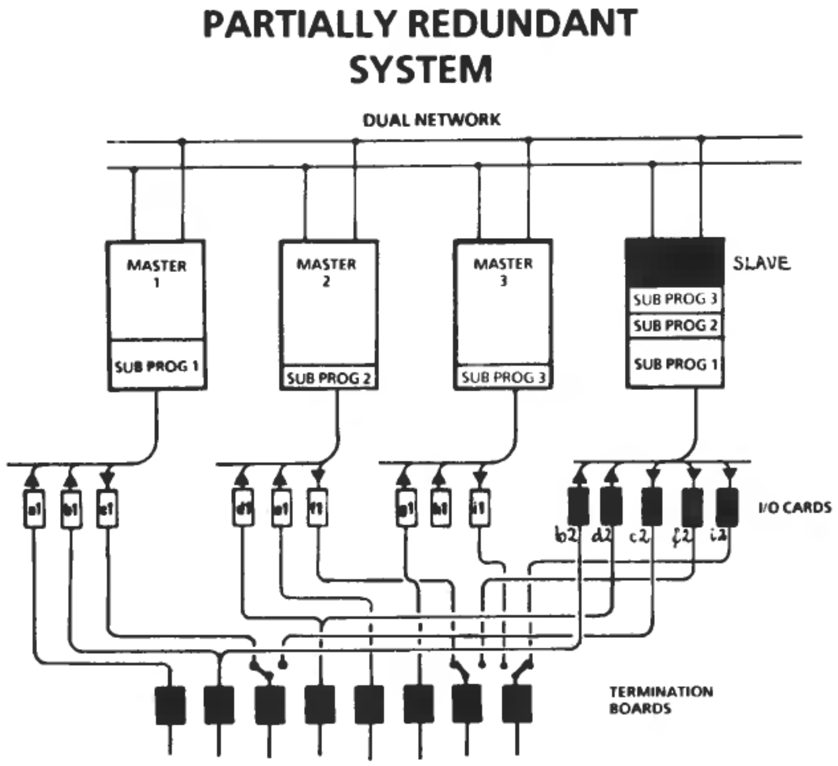


Figure 15. Partially redundant system.

In this system M1 and M2 are master computers and S1 is a slave computer. The slave computer is backing up F1 in M1 with the corresponding i/o cards b1, c1 and F2 in M2 with the corresponding i/o card f1. If M1 fails, the function F1 and the work of b1 and c1 are taken over by S1 and the cards b2 and c2. As M2 is still functioning, S1 is partially doing the backup function of F1 and partially acting as a hot standby for function F2 and the card f1.

6.5. Triple modular redundancy

The Triple Modular Redundancy offers an extreme degree of system availability. The TRM system uses three computers, which do the same task, simultaneously with input from the same sources via three parallel and identical sets of input cards. This system makes it possible to detect failures in the individual interface cards as well as in the computer-generated results.

The fault detection and isolation is done by using the majority voting concept. All three computers in a triad compare and vote on their critical inputs and results. The voting follows the 'two out of three' concept. Even if one computer module or an interface card should produce an error, the process will never see that error. The two processors which are in agreement will act as a majority and control the process. At the same time, the incorrect computer will, if possible, automatically correct itself, based on the values of the agreeing computers. If the failing computer fails to repair itself, the error is signalled to the operator and the faulty unit can be replaced while the other two are working. The new computer will automatically reboot itself as soon as it becomes active.

The output from each computer passes to its own set of output cards. A hardware based median selector (voter) selects the median signal from the three versions of each signal generated.

Triple Modular Redundant versions of a process station are avilable for operations under extreme reliability requirements.

6.6. Communication network redundancy

A dual redundant communication network will normally be delivered both at the management and the process control level. The two networks are electrically isolated from each other. In case of failure of one of the dual networks, the other will automatically take over. The passive network is continuously monitored and tested to detect faults in this passive network.

7. System generation and documentation tools

7.1. System generation

System generation can be made on-line or off-line. In the following it is assumed that the MMIf and the module library exists.

7.1.1. On-line configuration

On-line configuration is done directly on the target system. However, this is satisfactory only for very small and simple systems.

7.1.2. Off-line configuration

In most cases, the configuration is done in the off-line mode. This is presently carried out on a VAX computer, but any other computer with sufficient capacity and a C compiler can be used. An IBM PC/AT running the XENIX operative system could be used for example.

The configuration in the off-line mode proceeds in much the same way as in the on-line case. During the configuration, a simulator of the resulting AIM system is actually being built within the computer.

The configuration starts by interactively drawing the system flowsheet using the graphics facilities. When the system flowsheet is complete, a hardcopy may be produced by the computer.

At this stage different kinds of sorted module lists may be prodcued to help the user during the following work.

The next step is to configure the i/o systems in terms of numbers and types of i/o cards. A rudimentary i/o list can now be produced.

Assisted by the module lists, the system flowsheet, and the rudimentary i/o system list, the user can proceed by systematically going through the following configuration steps:

Detailed i/o system configuration. This can be done automatically or manually. In the automatic case, a file is supplied which contains a list of records, one for each signal. One record describes one i/o signal and includes the following items: A signal identification, terminal board connection, module tag name and terminal name to connect to, electrical signal type, physical signal type, scaling information, if relevant, and signal name.

The i/o file is automatically processed by the system which fills in the i/o system tables and connects the corresponding modules and terminals to the i/o system. If conflicts in signal types and connections occur, the system will report this and ask for corrections.

In the manual case, all these data elements are interactively typed in from the keyboard. Parametrization of the fixed modules. (Copy functions are available for duplication of parameter sets in identical modules).

Configuration of the configurable modules. This is done by drawing module diagrams, or fetching preconfigured modules from a library.

Configuration of the graphics presentation pictures and the reports.

During the configuration work, documents of different kinds may be produced at any time to support further work.

It is important to realize that the configuration, documentation, and control parts of the system constitute an integrated system. In order to utilize the true power of the system it should be used for more than just a control tool, it should also be a tool for planning, engineering and documentation of the control and instrumentation system.

7.2. System documentation

The system database is designed with a range of access possibilities. This makes the creation of different kinds of lists very easy.

The system supports the collection, sorting and printing of the following lists:

TAG LISTS. These may be sorted or selected based on type (function), process station, loops, process area or other criteria. The tag lists contain summary information about each module.

I/O LISTS. These lists may be sorted or selected in accordance with criteria such as sensor types, control valve types, electrical signal types, process station, etc. The lists contain i/o information like that described in the previous section.

ALARM LISTS. These are sorted or selected lists of alarms. Each alarm record contains information such as the associated tag and terminal name, a short description of the alarm, type of alarm (high, low, rate, switch etc.), type of function (alarm, start up or shut down), and physical alarm limit if the signal is analogue.

In addition to these lists, system flowsheet plots and module diagrams for configurable modules can be produced.

Detailed documentation of a module consists of its connection data, a parameter vector printout, and a module diagram plot (if the module is a configurable one). This complete module document can be produced for one or several modules.

8. Concluding remarks

This paper has discussed the design and implementation of a distributed control system and the configuration tools developed at Kongsberg Albatross a.s.

Besides giving the supplier or the user the possibility to implement new applications with minimum time and cost, the system gives the operator the opportunity to define his own working environment. As different access levels of operator configurations are possible, unauthorized use is prevented.

The extreme flexibility of the system allows the user to change, re-configure and expand his system while it is operating.

By acting as a simulator, AIM lets the user try out different control and man/machine interface strategies before the system is installed in the plant. This should reduce operator errors in the early phases of operation.

This system is facilitated by the use of the latest technology in hardware and software design. Much of the code written for this system is generated by the tools. The system is written in the C-language, and developed in a Unix operating system environment.

References

SAELID, S., 1986, Interactive modeling and simulation using distributed computers and graphical aids. *Modeling, Identification and Control*, **6**, 257–253.