CODEN: LUTFD2/(TFRT-5414)/1-75/(1989)

Designing a Process Operator Interface – a Hypermedia Model

Peter Höjerback

Department of Automatic Control
Lund Institute of Technology
October 1989

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Master Thesis

Date of issue
October 1989

Document Number
CODEN: LUTFD2/(TFRT-5414)/1-75/(1989)

Author(s)
Peter Höjerback

Supervisor
Nicholas Hoggard ABB, Karl-Erik Årzén

Sponsoring organisation
IT4

Title and subtitle
Designing a Process Operator Interface. - a Hypermedia Model.

Abstract

In the field of industrial process control the rapid growth of computer capacity has made it possible to replace the old and rather cumbersome control panels with new and user-friendly computer systems. The link between the computer and the operator - called the man-machine interface - will govern how quickly and accurately the operator can extract valid information and respond to it. Its design will thus be of great importance.

Within the national research programme IT4 future real-time knowledge-based control systems are studied. The goal of this Master Thesis has been to develop a model of a process operator's interface in this system. As prototyping environment a Macintosh (MacII) and a hypermedia program were chosen. For four purpose the German hypermedia program Plus proved best suited as it could handle both colour and fullsize screens.

The outcome has been a functioning system filled with colour, sounds and graphics as well as photographs and textual information. Each screen is linked to several others in a hierarchical organization allowing for rapid browsing and information zooming. Colours and icons are easily adjustable making often-desired individual adjustments possible.

The powerful tools of Plus and its HyperTalk-cloned object-oriented programming language have enabled a comparatively short development time. Being merely a pilot model the various features have in most cases only been exemplified to show what can be done.

The possibilities in hypermedia are countless and it is my belief that at least some of those engaged in system development, software evolution, artificial intelligence or man-machine communication could benefit from hypermedia models e.g. for rapid implementation of ideas and user influenced prototyping.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
75

Recipient's notes

Security classification

Table of Contents

Table of Contents (cont'd)

Acknowledgements

Developing the process operator interface I have continuously been discussing all aspects with my supervisor Nick Hoggard at ABB Corporate Research in Lund. His knowledge of computers at large and of man-machine communication basics have helped me a lot and he has most definitely improved the outcome of this Master Thesis.

Regular meetings have been scheduled with my supervisor Karl-Erik Årzén at the Department of Automatic Control at the Institute of Technology in Lund. Karl-Erik has introduced many bright ideas of how to design the interface.

For giving me the stimulation not only to do this very Master Thesis but also to begin my studies of Engineering Physics I want to thank Karl-Johan Åström, professor at the Department of Automatic Control at the Institute of Technology in Lund. His advice and thoughts of how I could organize the computer system and how to solve the question of "level-of-abstraction" were much appreciated.

Claes Rytoft, head of the ABB Corporate Research Department at the Research Park IDEON in Lund, has been closely involved in the interface development. His expertise on the Steritherm process has been very useful as has his experience regarding man-machine communication concepts. I especially appreciated his assistance in checking and re-checking this report.

Finally, I would like to thank the employees at ABB Corporate Research in Lund, Hans Pettersson at Syscon AB and Christer Gerding at SattControl AB for contributing with their opinions, listening to my various problems and helping me out.

1 Introduction

"There's a natural instinct to see either a revolution or a conspiracy in every new technology that comes down the pike"

-Russel Neuman [Brand, 1987]

This report is the documentation of a Master Thesis called "Designing a Process Operator Interface - a Hypermedia model". The Master Thesis is a part of a Master Degree in Engineering Physics at the Lund Institute of Technology and it was performed during the summer of 1989. Supervisors were Nick Hoggard and Claes Rytoft at ABB Corporate Research in Lund and Karl-Erik Årzén at the Department of Automatic Control at the Lund Institute of Technology.

The emphasis of this Master Thesis has been put into the development of the process operator's interface and the exploration of hypermedia as a prototyping environment. Colour and on-screen-action have been extensively used. Unfortunately this is extremely hard to describe properly in a short report. To compensate for this limitation full-colour screen copies have been integrated into in the original versions of the report.

1.1 Outline of the Report

This report guides you through the development and evaluation of a hypermedia application. The general conditions for the Master Thesis will be given in this chapter. In chapter 2, called "Development Conditions", some of the different software and hardware will be described.

Chapter 3 is called "Man-Machine Communication" and turning the pages over you will be given the reasons why a process operator's interface should look a certain way and how this is achieved. You can then study how a "three-dimensional" picture or a technician's flow chart are developed in chapter 4 "Working with the Hypermedia Program Plus".

Tying the loose ends, screens and knowledge-bases together in chapter 5 "A Look at the Resulting Interface" a small system emerges. It will show you how a process operator's interface in a knowledge-based real-time control system could look in a couple of years. The same chapter also tells you of some ideas not yet implemented in my model and of possible improvements.

In the last chapter called "Conclusions" you will accordingly find a collection of my experiences. There is also a paragraph about what new features that could be offered in the future. The final appendices contain additional information on processes in common (A) and the process operator views (B).

1.2 Computers in Control Systems

The rapid development in the world of computers provide us today with the possibility to improve and simplify many tasks. One of them has been the writing of letters and memos nowadays easily done on a word processor.

A totally different and more complex task is to control an industrial process. It could for instance be to sterilize milk, pour milk into Tetra Pak packages or to make paper out of pulp. The use of computers in control systems is nothing new, but the extent of the use has been limited. Today, a lot of new things can be done with the refinement of knowledge-based systems and the use of artificial intelligence on the software side and the introduction of more price/performance effective computers with graphical processors on the hardware side.

The amount of information handled by such systems and the comparatively small area on which to present it (say 19" or 35 x 26 cm and one to three screens) put specific demands on the link between the computer and the process operator. How well this link has been designed determines how quickly and accurately the operator can extract valid information and respond to it.

The design of this operator's tool will further be a key factor to the developing company's success as it is the "face" of the product. But the design of the system is important also from another aspect - the operator's personal satisfaction.

Many changes are brought about with the introduction of computerized control systems and previously highly valued skills gained after years of experience suddenly become something computers can do as well. Sometimes these jobs get so automatized that the skilled operators feel that this has dehumanized their work [Ehn & Kyng, 1984].

This phenomenon is even more topical today as artificial intelligence can come to conclusions and make decisions without the operator's help. A good design of the system and its interface will minimize these feelings and improve the operator's personal satisfaction. All together we can understand why man-machine communication is an important subject.

1.3 IT 4 and Knowledge-Based Real-Time Control Systems

Let us now take a look at what industrial computers currently are doing. Let's study some typical computer programs for accounting, calculations or process control. We will find that it is in fact only repetitive and algorithmic functions like arithmetics, sorting and merging that prevail. However, the improvements made in these areas have been enormous.

Still there are a lot of things computers haven't been able to deal with in the way we would like. One such thing is "knowledge", which by all means is included in the application programs of every computer system, but which computers haven't been able to store explicitly. This means that if you want to control a valve in a current computerized system you would have to read through all the program code as the "knowledge" of this valve is found in many different places. Instead of this tiring work, you would like to use a knowledge-base and have all the "knowledge" in one place.

This is where new programming technologies enter the scene. Object oriented programming, knowledge-based system techniques and applied artificial intelligence will in the future make it possible to represent and process knowledge. It will also be possible to store it in a single location.

These thoughts form the basis of a joint Swedish research project between Asea Brown Boveri AB, SattControl AB and the Department of Automatic Control at the Lund Institute of Technology. The project is called "Knowledge-Based Real-Time Control Systems" and it is performed within the national IT 4 programme. IT 4 is short for Information Technology, research programme number 4, and is partly government sponsored.

The main attention of the project is focused on the studies and development of knowledge-based real-time control systems for industrial processes. With a process is meant a controlled flow of matter, energy or information from generation via transport, storage, distribution and change to consumption (DIN, 1985). Such processes are found in the traditional process industry, the manufacturing industry and in telecommunication systems. See appendix A for the details of why this is true.

To include all knowledge about a process in a computer is an extensive task. The following would have to be considered:

- the controlled process,
- the involved components and how they interact,
- the produced products and their properties,
- control theories,
- demands and conditions for service and maintenance etc.

Further on, the knowledge of the designers should be included because they are the ones who have done investigations and know why a certain component has been chosen. Accumulated knowledge of the process computer and the operations personnel, gathered during years of operation, would also be most valuable. What kind of system can then be envisioned?

1.4 The Visionary Goal

As an answer to the above question a concept to a new system has emerged from the IT 4 project group. It is mainly based on the belief that knowledge representation and use cannot be separated [ABB AB et al, 1988]. That is, it must be integrated into one entity, symbolically depicted as a circle with the knowledge representation in the middle and the users and their tools surrounding it (fig 1.4.1).

Knowledge about the process will be stored in a common knowledge-base forming the heart of the system. The structure will probably be of a hierarchical and object-oriented nature. There will of course also be a real-time link to the process itself.

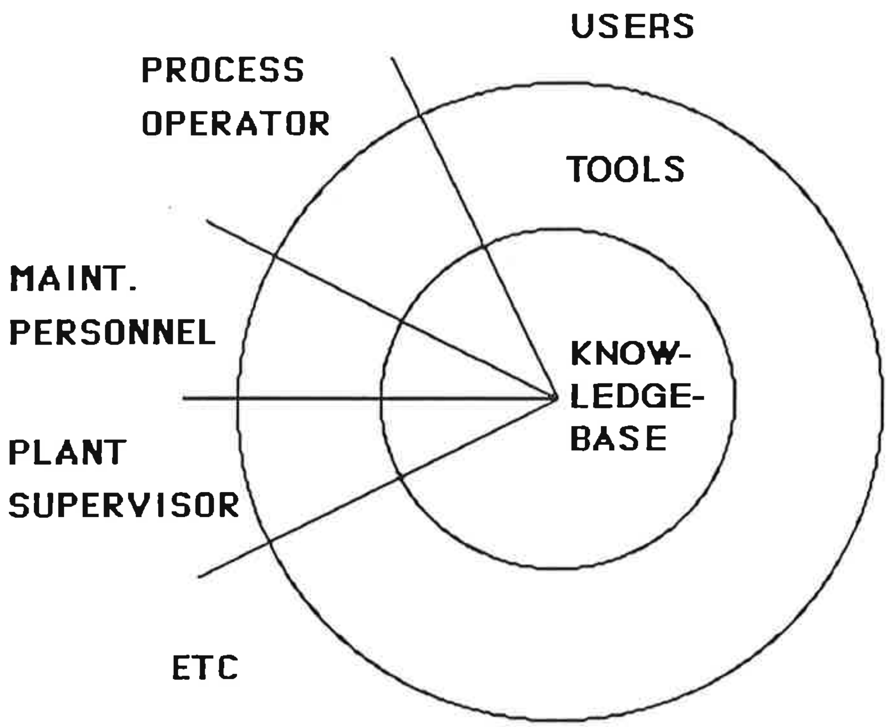


Fig 1.4.1 The system concept of the IT 4 project group.

To operate on the knowledge-base and control the process you will be given a set of tools. Each set is individually tailored according to what job you are performing. If you are responsible for maintenance you would need to know the state of sensitive mechanical components and the stress history for that component. Has everything been normal or do you have to take preventive maintenance actions like changing a valve ahead of schedule? Do you have any spare valves in store or must new ones be purchased?

No matter how important these questions are to the maintenance staff, a process operator or a designer would very seldom have any reason to ask them. They have other concerns like if the product has been thoroughly treated or if the design of the heat exchangers is optimal. Each user will thus need an individual set of tools. If these tools are to be of any meaning they must somehow improve the job of the user. The users must feel in control of their new powerful tools and yet be able to get both help and assistance.

For this purpose you need a well designed human computer interface (see chapter 3 for more on human-computer interaction and man-machine communication). Each user will look upon such an unique interface from which he will be able to benefit from all the "goodies" of modern technology.

It is here my Master Thesis enters. Its goal is to emulate (pretend to be) the interface of the process operator of such a knowledge-based real-time control system as thoroughly as possible. A multitude of aspects will be illustrated and on selected items implemented in depth. There will for instance be zooming possibilities only on a few pieces of equipment, but very detailed on those ones. Up-to-date methods on how to design good man-machine communication will be used (as far as I know them...) and some new concepts such as how to use sounds will be tried out.

This all leads us to the visionary goal of IT 4. That is a control system that could be programmed in a knowledge and problem oriented way and with which you could "have a conversation". In fact, just as if you had asked the designer of your system or the person who had the process operator job before you. It is the belief of the IT 4 project group that correctly made such a system could, among other things, improve the time to master a complex system. A small step towards this goal has been taken with hypermedia techniques as they allow build-in instruction books and manuals and the incorporation of new medias like video films, photographs and pictures.

1.5 What is Hypermedia?

Hypermedia is more than just another new word in the inventive computer business. It signifies a new way to deliver information, in whatever form it may be, that goes beyond computer lists and database reports. Hypermedia puts all information into an eternally cross-referenced web. In this web you can wander around freely and find your piece of information whenever you need it.

Let me give you an example of what I mean. Suppose you are a law student interested in the laws restricting insider affairs at the Swedish Stock Exchange Market. Having studied the Book of Laws you now want to find the most recent trial cases. A hypermedia link will connect you to it, although it is in a completely different reference work.

The accused company could then be studied further by another hypermedia link taking you to a National Trade Register or to the phone book where rapid search functions will help you find the information you are looking for. By clicking on the company's number the computer will automatically dial the number for you. Video films could also be shown about the company if such were available. "We can focus more on content, while ignoring the organization." as John Scully, Chairman and CEO of Apple Computers, puts it [D Goodman, 1987].

Hypermedia is in fact not a concept of the late 80's. Its origin can be traced back to visions of macro literary systems in the mid 40's! The vision was to integrate colossal volumes of information to make them readily accessible via a simple and consistent interface. During the last twenty years the word **hypertext** has been used quite frequently, but loosly, meaning a lot of different things. To make things easy you can say that hypertext is hypermedia without digitized speech, audio recordings, pictures, photographies and animation; just the textual information and the brilliant linking and search functions.

The break-through for hypermedia didn't come until 1987 when Bill Atkinson presented the first commercially successful hypermedia program. He called it **HyperCard**. Until then personal computers hadn't had the capacity needed to handle a hypermedia application. The amount of information would simply have been to large. Nowadays it is both possible and affordable.

As high-capacity magnetic hard disks are standard on most computers today we have access to very large amounts of information. A common hard disk of 40 megabytes equals to 20 000 pages filled with text! But that's not very much, currently CD-ROM's, optical hard disks and other mass storage technologies are entering the market and they are getting more and more affordable. It is here the new hypermedia programs show off.

Their capabilities to rapidly search for a certain piece of information and then copy or use it are very attractive. Likewise are the MacPaint-like graphics and the way you can control both the execution of your program and exterior devices (like a TV-set, a laser disc or a sterilizing process machine).

1.6 Steritherm - the Project's Guinea-Pig

If you are to develop a research prototype of an operator's interface in a knowledge-based real-time control system you want to be sure that the problems you encounter in the demonstrator, i.e. the actual process you are studying, have general applicability. The important thing is to find new solutions to common process problems like how to analyze an alarm or how to include and represent design knowledge in the system.

To find such a representative process is possible as many modern and complex processes have a number of common features [Dhaliwal, 1985]. Among them are the inability for a single individual to fully understand the process due to its complexity, the tens of volumes of operations and maintenance manuals and the rapid evolution and changes to the process. They also share a common need for speedy and accurate corrections of faults (see Appendix A for more details).

An evaluation done by the IT 4 project group [Rosenberg et al, 1989] shows that Steritherm, Alfa-Laval's full-scale process for indirect ultra-high temperature (UHT) sterilization is well suited as a demonstrator (fig 1.6.1). Although it is a relatively simple process it contains several non-trivial problems. I will explain the process to sterilize milk.

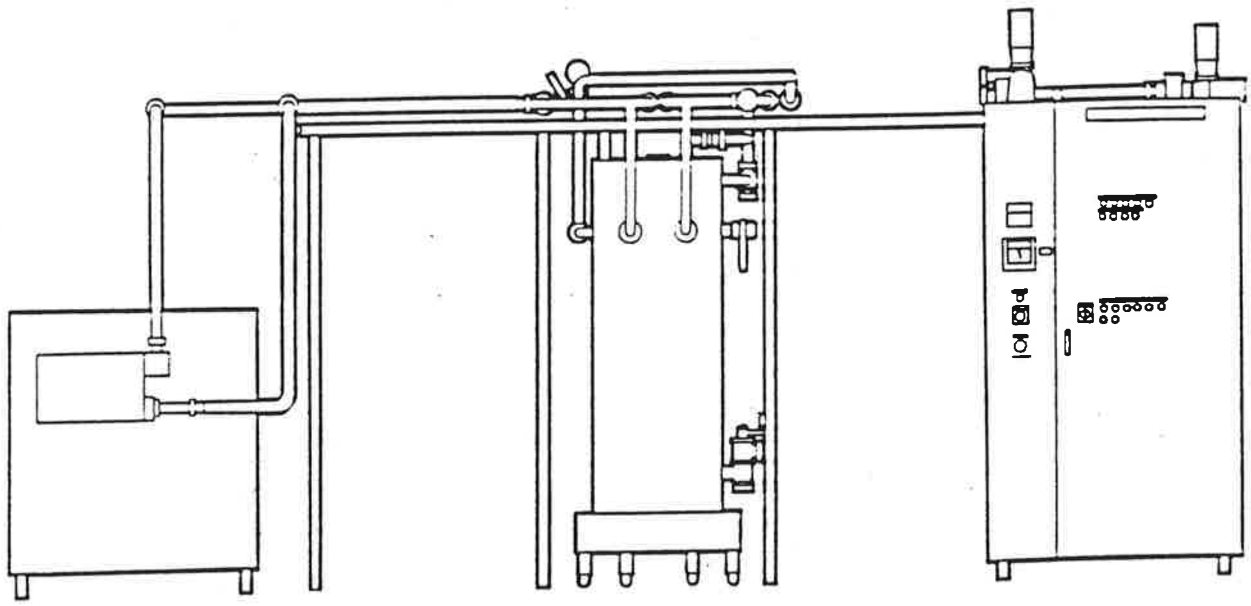


Fig 1.6.1 The Steritherm sterilization process machine.

First you have to make sure that the process equipment is sterile by initiating the **sterilization** sequence using hot water. You then continue to **production**. Pumping the cold and fresh milk by way of a composite heat exchanging arrangement gives it a temperature of about 137°C. This temperature is kept for 4 seconds thereby giving you milk free from all micro-organisms, which otherwise would have grown and contributed to deterioration (=sterile milk).

The cooling of the milk is done by the same heat exchanging arrangement and this time you use the hot milk to preheat the incoming cold and unsterilized milk. The sterile milk is finally led to an aseptic packaging machine and sealed. Your sterilized milk can now be stored for months or longer. More information on dairies and milk can be found in the Dairy Handbook by Alfa-Laval [Alfa-Laval].

2 Development Conditions

"The medium is the message."

-Marshall McLuhan [Brand, 1987]

2.1 The Hypermedia Software - a First Evaluation

One of the objectives of this Master Thesis was to evaluate hypermedia as a prototyping environment. A survey of the hypermedia programs on the software market led first of all to HyperCard. Its commercial success has made it widely spread, its reputation is good and several handbooks are available. Among them is the excellent and humorously written "The Complete HyperCard Handbook" [D Goodman, 1987], which has been very useful. There is even a Swedish translation available [D Goodman, 1989] with updating on the latest version called HyperCard1.2.

As I began my work I had no practical knowledge of HyperCard at all, but some experience of the Macintosh. I was also familiar with computer programming at large and of a few computer languages like Pascal and Fortran. Fearlessly and with the handbook and a manual next to the computer a journey into HyperCard began. It went surprisingly smoothly and I was writing my own small programs after only a few hours of "browsing around". During the following days the handbook was frequently used, but the manual was hardly touched. It simply wasn't written with the glint in the eye as was the handbook. There are also a beneficial introduction and an extensive help system integrated into HyperCard.

Fig 2.1.1 The icon for HyperCard.

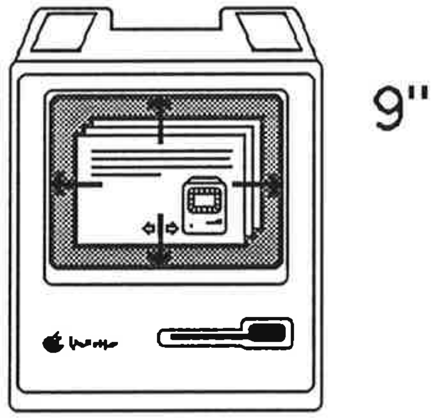


Fig 2.1.2 The ordinary 9" Macintosh screen.

No matter how fun it was to work with HyperCard, it had its limitations. Using a full-size (13") colour screen it felt like a waste to only be supported with grey scales and a window the size of an ordinary Macintosh (9"). Besides, colour and the use of a large screen are two of the best ways to make computers and humans communicate efficiently. That disqualified HyperCard already at an early stage from being chosen.

At least two hypermedia programs remained, both claiming to provide colour and full-screen support. They were **Plus** from the German company Format Software GmbH and **SuperCard** from Silicon Beach, California. In spite of several contacts with both Swedish suppliers and Silicon Beach no date could be given when a copy of SuperCard could be delivered. Not even a prototype was available at the end of May, 1989. Thus SuperCard had to be eliminated.

While all this happened Plus arrived on time in my mailbox. To work with Plus was a very different experience from HyperCard. All of a sudden the **whole screen** was filled with **colour** graphics! Nice indeed! Where HyperCard felt inadequate or restricting Plus offered extensions and new features. I will give you an example.

Fig 2.1.3 A typical button from HyperCard or Plus.

In hypermedia one of the basic building blocks are buttons, mouse-sensitive areas that execute a program when clicked on (fig 2.1.3). In HyperCard you can only have rectangular buttons. Still, on a process operator's interface you may very well want circular or totally irregularly shaped buttons. It can for instance be a pump you want to depict. In Plus you can make irregularly shaped buttons, but they are then called paint objects. The secret of paint objects is that they are only sensitive to clicking on filled pixels (fig 2.1.4) and ergo you can have any button-shape you like.

Both HyperCard and Plus offer excellent possibilities to store and display text. HyperCard is however limited to one single textual style in a given text field whereas Plus will give you full word processing capabilities. Such fields permit the user to employ a number of fonts and they permit full flexibility in terms of font sizes. You can even colour your text the way you want and then edit it as usual! To be quite honest HyperCard does offer a possibility to use whatever style you like. This text will be regarded as graphics and is thus uneditable and unscrollable and of course without colour.

Fig 2.1.4 An irregularly shaped paint object from Plus.

Student of Engineering Physics
Peter Höjerback
Lund Institute of Technology
Box 118
Ole Römers v 1
S-221 00 Lund
Sweden

Fig 2.1.5 A text field from HyperCard.

The price you have to pay to get the extra features of Plus (apart from the price tag at 3000 SEK) are inferior documentation and a slight feeling of instability. The major drawback compared to HyperCard, I think, is the bad documentation. Directing your call for help to the attached manual won't do much good as you'll discover that it is almost impossible to find what you are looking for. The built-in help function is unfortunately neither user-friendly nor self-explanatory and you seldom get much wiser by using it. Luckily, many functions are found in "The Complete HyperCard Handbook" [D Goodman, 1987], already mentioned above. "Learning by doing" is otherwise a recommendable method in Plus...

Finally Plus hasn't got the same stability and genuine feeling of quality as HyperCard. This could be explained by Plus's recent arrival on the market (spring 1989) and that it hasn't been as thoroughly tested as HyperCard. Besides, we're comparing with the adjusted second version (1.2) of HyperCard.

Summing up advantages and drawbacks the colour and full-screen capabilities of **Plus** make it the **best choice** - at the time being - for this application.

2.2 Auxiliary Software

Based on a survey Apple has made, the company claims in its marketing that Macintosh users run several different programs each day and even more than an average PC-user. Many different programs have been tried out during the development of the interface and in this chapter some of them will be presented. The hypermedia programs have been described separately in the previous chapter 2.1.

Graphics & Scanning

It is interesting to know if it is possible to draw pictures in **MacDrawII** and then import or export this kind of graphics (PICT) to a hypermedia application. You can in fig 2.2.1 study a simple MacDrawII picture that without any problem could be moved from MacDrawII to Plus. It is my personal opinion that MacDrawII was really easy to learn and fun to work with.

Although most pictures were drawn inside Plus, the most beautiful ones have been scanned with **ThunderScan** and then imported into Plus. How such a scan looks you can see in fig 2.2.2. Once inside Plus they have been edited and coloured with the graphic tools to look a little better. Additional ways of processing a picture were available in **DigitalDarkroom.** Its ability to change and improve a picture are amazing. It would however be beyond the scope of this work to spend hours perfecting pictures.

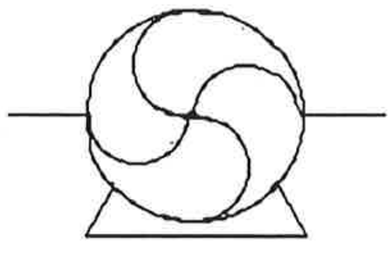


Fig 2.2.1 A very simple MacDrawII picture showing a flow chart symbol of a pump.

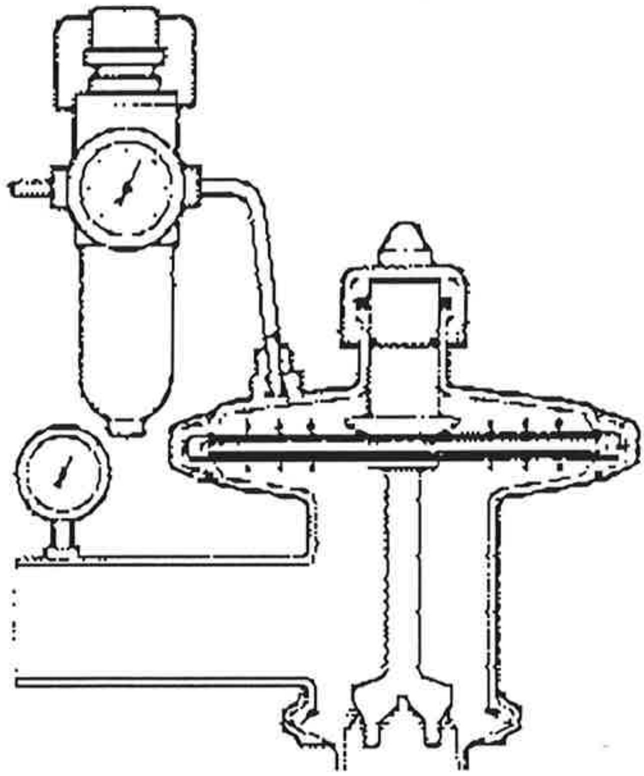


Fig 2.2.2 A scanned picture showing a valve (ThunderScan).

Sounds

To have a direct verbal conversation with your computer may not be so far away any more, but as of now you will have to settle for prerecorded messages and sounds from a sound generator. These sounds can be excellent, even of CD stereo quality. Adding sounds will mean something new as the process operator's interface of today is almost without them. This situation is about to change, at least in environments that aren't too noisy.

With products like the INIT-program **SoundMaster** several interesting sounds can be generated like BLIRRRP! when you enter a diskette or the theme from "Star Trek" when you turn on the computer. Another program called **Sound->snd** translates a file with a digitally recorded sound into a sound resource.

A resource is a small attribute like a sound, an icon or a menu that is located outside the ordinary program code. This has the advantage that you can easily change an icon or modify a menu without having to recompile the whole program, which can be really time-consuming [D Goodman, 1989].

Fig 2.2.3 A friendly voice.

If you want to include a certain sound in your hypermedia program you will have to move the sound resource there. It is easily done with a resource editor and I used one called **ResEdit**. ResEdit works just like the Macintosh Font/DA Mover [Apple Computer, 1987] with simple clicking and Copy & Paste.

You can now have sounds like Clint Eastwood hissing "Go ahead, make my day.." when an operator wants to delete something or, more seriously, a friendly voice telling you "It is time to clean the pipes. We have been producing for eight hours." if that is the case.

Word Processing

Microsoft Word is a well documented standard word processor used by all the secretaries at ABB Corporate Research. To use it for the writing of this report could be quite suitable. Numerous advanced features and short-cuts are provided and some of them are useful even to a less experienced user like myself. In some cases, like the first time you use the outline function, you will have to spend both time and effort to learn it, but you will hopefully get it back in the end. After only a few hours of writing I decided to finish the report on MS Word and so far I haven't had any reasons to regret that choice.

2.3 Hardware

All the work and demonstrations have been done on **Macintosh II:s** with 5 MB expanded primary memories and **40 MB hard disks**. It was possible to admire - and bewail for that matter - the results on **13" Apple High-Resolution Colour Monitors** and on paper printed by the **LaserWriter Plus**.

For scanning pictures a small **ThunderScan** device mounted on an old ImageWriter was used. It didn't go very fast but it did what it was supposed to do.

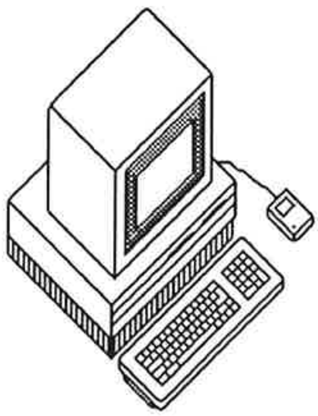
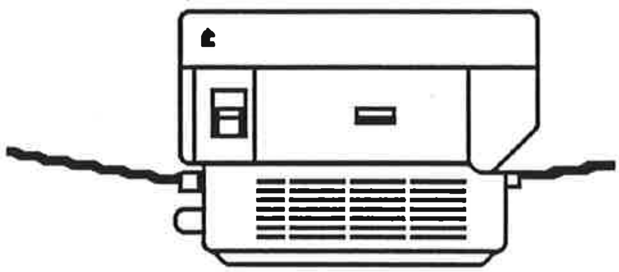


Fig 2.3.1 The Macintosh II.

As the size of the interface file grew larger and larger a back-up problem occurred. One day the file was well above the 773 kB you can tuck into a diskette. At least two solutions were at hand. One was to split the file into two parts using the application program "Safety Copying" and the other was to store the file somewhere else than on diskettes. The last option was the most simple and elegant as I merely connected to a **Hewlett Packard 9000/300** via an AppleShare network and NFS - a standard network file system - and put a copy on its 300 MB hard disk.



2.3.2 The Apple LaserWriter Plus.

3 Man-Machine Communication for a Process Operator

"How will we directly connect our nervous system to the global computer?"

-Rory Donaldson [Brand, 1987]

This may very well be the basic question of a science called man-machine communication. Its objective is to make the interface between a human being and a computer as good as possible. Having been aimed at making humans to behave as machines during the 60's, the last twenty years' development has instead been concentrated on **humanizing computers**. By this I don't mean that computers actually have been built to behave like human beings, but that they have been made to be easily and naturally handled by human beings. Compare it to a car; you don't try to make the car look and act like a human being, but you make it "friendly" towards human beings. This approach has been far more successful. Today, anybody familiar with e.g. typing can benefit from powerful word processing computers.

This chapter will focus on the needs of a process operator regarding man-machine communication. While presenting some concepts of how to design an interface fulfilling those needs some examples from my model will time be shown in parallel.

3.1 What You Want

When you as an operator are running a process you want to do it as efficiently as possible (at least your boss hopes so). Working from a computerized control system you want to feel competent in using the system. You should easily be able to accomplish your tasks and errors should be just as easy to correct.

You would probably want to do a good job (-"As I'm here I might as well do the best I can.") and feel satisfied when you leave for home with what you have done. You would further want to feel that your skills continuously are being used and developed. Shouldn't it suffice you would like to get support both from your system and your colleagues. By also providing you with the necessary training and education the quality of your job would be assured.

3.2 How To Get It

The above stated thoughts seem plausible - or even obvious - but if you are studying computer systems you will discover that designers haven't always thought so. How could you as a designer of human-computer interfaces do things a little better? In the following I will present some factors that could be worth considering. I will also tell you what I have done in my Plus program to improve those factors.

Time to learn - the system must be predictable. This is the case if the operator is the initiator of actions rather than a passive responder. He will then feel in control and it will help him learning to master the system. Hypermedia programs are well suited from this point of view due to their object-oriented structure. If you click on an object a certain program called a script is executed. Often a message is passed to another object and something happens. You can for instance be taken to another view.

In these programs nothing will happen by itself - except if you have programmed it. This is of course not true in a real control system, which is dynamic. The only time this happens in my model is when you are simulating an alarm. The operator will then have to respond, survey the situation and eventually initiate proper counter-measures.

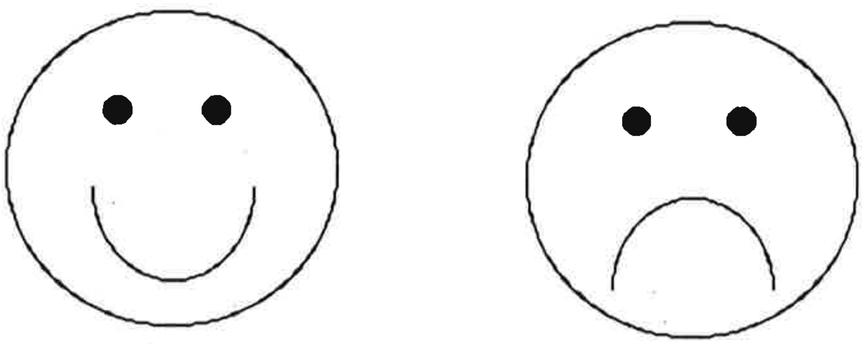


Fig 3.2.1 A happy and a sad Chernoff's face.

The "time-to-learn" will also be enhanced by consistent use of colour and icons (small symbols with a special meaning, just like a traffic stop sign telling you to stop). Icons, in the shape of buttons or paint objects (chapter 4.2), are two of the basic building blocks in many hypermedia programs. Myself, I have used them as much as possible eliminating the need for a keyboard when running the process. If you e.g. want to go to the Main View just click on the icon with the happy Chernoff's face and you'll be there. Chernoff was a man who invented a whole scale of facial expressions telling you symbolically of feelings like happiness, sadness, bewilderment, anger and so forth.

Speed of performance - the importance of quickly providing the operator with valid information and support rapid decision-making. All systems should present the information in a way that simplifies decision-making. If this shall be obtained one of the most crucial things is to restrict the amount of information being forwarded to the operator.

One method to solve this problem is to introduce different levels of abstraction. If the process is doing just fine and nothing calls for the operator's attention the Main View is shown. It consists solely of a happy Chernoff's face and some icons to take you to the most important screens.

Suppose now that the operator wants to control the sequence being run, e.g. a sterilization sequence. He then clicks on one icon and he will be at an operations screen giving him general sequence information. By clicking once more - this time on the sterilization name box - he will zoom into a screen showing the sterilization flow chart with real time process values of temperature, pressure, etc. Panning and zooming in this way lets the operator choose at what level of abstraction he is studying the process and how detailed information he wants. The organization of the information is in this way hierarchical.

By making it possible to control the process from a high level of abstraction the operator will usually not have to endure time-consuming shifts of views and an overwhelming amount of information. Instead he will instantly be able to start the next sequence from his overview screen.

Rate of errors - make the system simple and be sure that it is easy for the user to correct his errors. Self-induced errors are indeed a common source of nuisance to all computer users. The rate of errors demands special attention as it can significantly reduce the speed of performance. On-site studies of the most frequent operator errors have in many cases proved valuable. Such studies could very well be performed on prototypes developed in a hypermedia environment and you will be saved a lot of tiring and costly adjustments of the final interface [Luqi,1989].

Interactive modifications with the presumed user towards a forgiving system will always be appreciated, but being merely a model of a process operator's interface I have not found the time to perform "live" testing, although it would be extremely interesting.

Subjective satisfaction - ensure proper feed-back and a likeable system. I think that a key factor to satisfaction is to custom-build the system and adapt it to the individual user. The operator should be able to get the colour coding of temperatures that he prefers (he may even be colour-blind and vote for patterns instead of colours) and the kind of parameter displays that he feels are most easily read. The examples of this kind of details are countless.

To have a choice of which sounds that are most appropriate, for instance when zooming, could also be a detail making an operator feel more comfortable or even more responsible. However, a contradictory desire is to have all operators using a standardized interface. The purpose for this is to eliminate the risk of confusion, especially during emergency situations when many operators could be looking at the same screen. A trade-off between these two goals has to be made.

Individual changes like these are easily made in hypermedia programs. Of course many other factors contribute to personal satisfaction ranging from purely psychological factors like feed-back from colleagues to education and training possibilities and access to short-cuts for experienced users.

Retention over time - do you remember how to operate the process after having spent five beautiful summer weeks in a hammock? The question is closely linked to the "time-to-learn" and to how consistent the design of the interface has been made. How much time you have spent together with the system will also be important.

Your memory of how to get to a card is continuously reinforced by the designation of a representative icon to each card. Whenever you want to go to a certain card just locate the icon and click on it. Zap! and you are there. For additional reinforcement the card's own icon is also present near the upper right-hand corner.'

References to some of these ideas and thoughts have come from [Schneiderman, 1987], [Norman & Draper, 1986] and [Gertman et al, 1985].

4 Working with the Hypermedia Program Plus

"Designing an object to be simple and clear takes at least twice as long as the usual way. It requires concentration at the outset on how a clear and simple system would work, followed by the steps required to make it come out that way - steps which are often much harder and more complex than the ordinary ones. It also requires relentless pursuit of that simplicity even when obstacles appear which would seem to stand in the way of that simplicity."

- T. H. Nelson [Nelson, 1977]

In this chapter I will tell you about what it was like to work with Plus and how I went about strictly practically to design a screen. As I gradually gathered experience I came to develop new and more efficient design methods. Screens of an early design will thus be somewhat different from those more recently done. A process flow chart will be chosen as an example of a complex and time-consuming screen of an early date. A newer and more sophisticated design will be a "three-dimensional" picture of the Steritherm process. To make you feel a little bit more at home when reading this chapter, it begins with a short introduction to the system followed by a presentation of useful expressions in Plus.

4.1 A Brief Introduction to the System

Imagine that you are a process operator sitting in front of a computer screen. Your job is to run a sterilizing process machine - the Steritherm. You will be able to control it with single mouse clicks on sensitive areas on the screen.

At your disposal you will have a system, whose organization can be regarded as hierarchical. On the top you will find the Main View, which is also the highest level of abstraction. You can't see any drawings or parameter values, like pressure and temperature. Instead the state - happy or sad - of the whole process is displayed using a Chernoff's face. From here you can immediately reach the Operations View, the Information View, the Economics View and the Knowledge-Base View (fig 4.1.1).

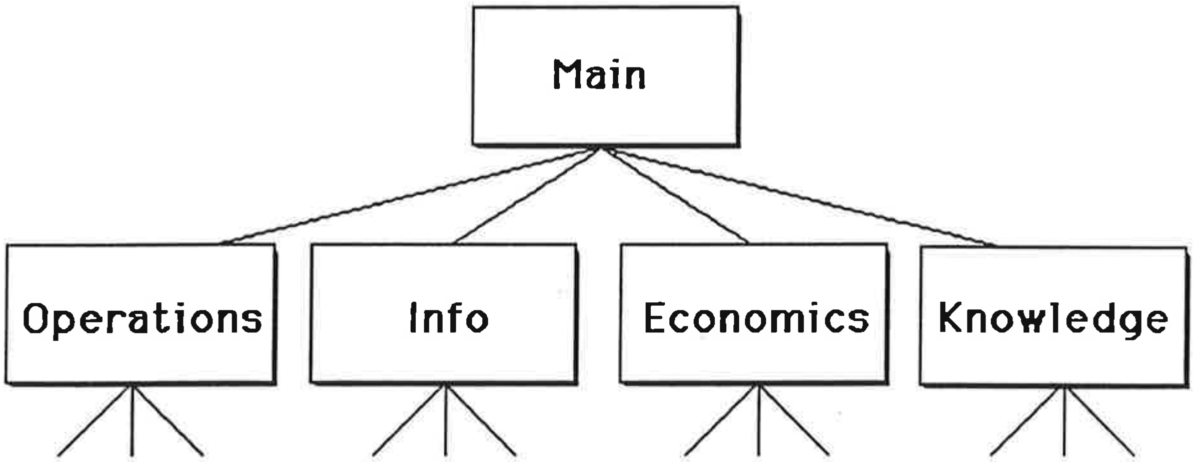


Fig 4.1.1 The organization of the system.

The name of a view gives a hint to the purpose and functions of that view. Accordingly you operate the process and give sequence instructions like 'Start sterilization' or 'Stop production' from the Operations View. In the same way you will from the Information View have access to all kinds of information on the process ranging from the smallest screw to a complete flow chart of the whole process.

Clicking once one the Economics View icon you will be shown a parameter recording of how efficiently the process is being run. Is it at a maximum, or could you perhaps produce more? Had you instead chosen the Knowledge-Base View you would have had an expert system at your hands. The thought is (meaning that I haven't implemented it) that from this screen you should have access to the common knowledge-base containing the underlying process and control knowledge.

Apart from these special functions each card contains several icons linking you to more information and to other views. Multiple ways of locating the desired information are present and the representation will vary from ordinary line drawings to photographs and advanced perspective pictures. Colours and sounds will facilitate your search. A more exhaustive presentation of the views can be found in chapter 5.1.

4.2 Stacks, Scripts, Buttons and Other Peculiar Expressions

The vocabulary of Apple's HyperCard is very similar to ordinary straight-forward English. The same goes for its programming language HyperTalk. Being the commercially so far most successful hypermedia program HyperCard's vocabulary has almost set a standard. This is especially true in Plus, which you in many ways can consider as an extension of HyperCard. Before I begin to tell you about what I have done I would like to explain a few strange words and their meaning.

The five first words are the basic building blocks of both HyperCard and Plus:

-Stacks
-Cards
-Backgrounds
-Fields
-Buttons.

Plus alone lets you add:

-Paint objects
-Draw objects
-Word processing fields

Stacks. A stack is the largest building block and it is usually a collection of information stored in one Macintosh file. You can think of a stack as a drawer in a card catalogue. If you double-click on its icon in the Finder you will open the drawer and have access to all the information.

Cards. In your drawer all the information will be stored on cards. What is found on one card is unique and can usually not be found on any other. The subject of the cards in the drawer is however the same. You could for instance label your stack "Cars" and all cards would hopefully be related to cars (if you prefer a logical approach to your labeling...). The individual cards could then be labeled "Volvo", "Saab", "Chevrolet" or whatever brand you like.

Background. On all the cards you would probably want notes on horsepower, speed, year of construction and so on. The space where these notes are put will perhaps be small boxes with titles like "Speed" in front of them. As you would like to have this information on all cars you put the boxes and the titles on a background, which is common for all (in general at least) the cards in the stack. Right now you are thanking your lucky star that you didn't have to draw hundreds upon hundreds of lines just because you like cars and want to organize them.

Fields. A field is where you enter textual information. You may have another kind of text on the background saying "Car Register", but it is not likely that this is the text you are browsing through the stack for. Instead it is the text specific for a Chevrolet Corvette 1989 you want. The text in a field will thus usually change from card to card. Perhaps some cards don't even need the presence of the field. The choice is yours.

Buttons. A button is a rectangular icon of your own design with a very special property. If you click on it a small program will be executed and some kind of action will take place. Looking at the card of the Chevrolet you could have a button called "Cross-section" and another called "Photo". Clicking on the first would show you a new card with a section drawing of the car - no need to go to another office and in their card catalogue try to find the cross-section as is usual today. In the same way you could study a photo of the cars although it was located in a completely different place. Buttons are as you perhaps begin to realize the primary navigational tool in these vast seas of information.

Paint objects. A new feature of Plus', paint objects can be compared to flexible buttons. As I have already mentioned, buttons are excellent tools although they suffer from the limitation that they have to be rectangular. In Plus you can use a paint object when you want a mouse click sensitive area of an irregular shape. This is because paint objects are only sensitive on filled pixels. I have found this feature extremely useful and have hardly had any need for the original buttons.

Draw objects. Plus provides a possibility to use simple object graphics. You can for instance draw a circle and afterwards treat it like a circle and not like a bunch of individual circularly positioned pixels. In my application I have never had any need for draw objects as they were difficult to handle and couldn't be grouped.

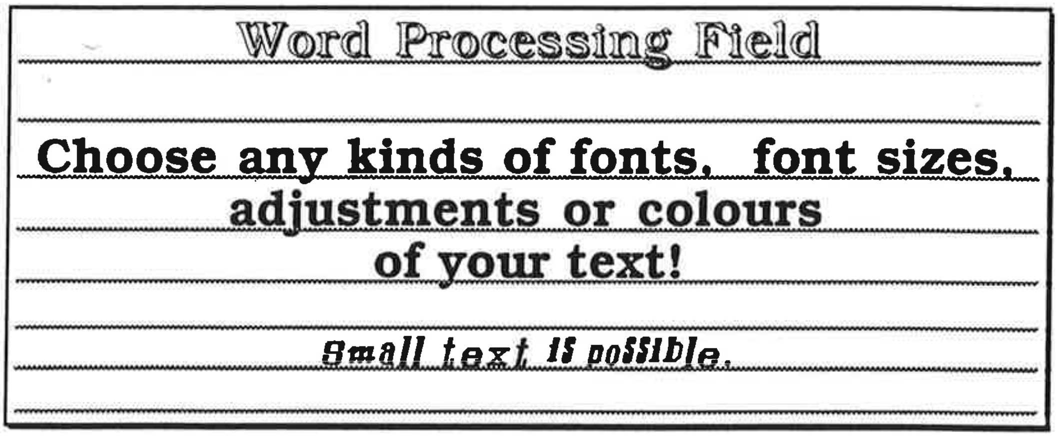


Fig 4.2.1 A word processing field in Plus.

Word processing fields. An ordinary field will only allow one style and one font of the text, which is not quite ideal. In Plus you can stop using ordinary fields and begin with word processing fields as they don't have this limitation (fig 4.2.1). They permit any mixture of styles and fonts and even colours. Good indeed.

```
script of bkgnd button "gotoinfo"                89-08-30    Page: 1

on mouseUp
 visual effect iris open very slow
 go to card id 8105
end mouseUp
```

Fig 4.2.2 An example of a HyperTalk-cloned script.

Script. A script is a program written in HyperTalk or any of its cloned extension languages. More often than not it is just a few lines as in the example in fig 4.2.2, but it can also be several hundreds of lines long. Another script, this one from the WHY??-button of the expert system, is shown on the next page.

```
on mouseUp
 if not the visible of background paintobject "hidealarm" then
  beep
  answer "Acknowledge the alarm first!" with "Cancel" or "OK"  at 140, 60
  exit mouseUp
 else
  play "Sorry Dave"
  answer  "Sorry, no help available yet...Click to reset the alarm!" with "OK" at
  140,60

  hide background paintobject "WHY??"
  set the cursor to watch
  set the lockScreen to true

  push card
  go to card id 3570                        --Main View
  hide card paintobject "sadFace"
  show card paintobject "happyFace"

  go to card id 8793                        --Operations View
  hide card paintobject "motorAlarm"
  pop card

  set the lockScreen to false
  set the cursor to hand
 end if
end mouseUp
```

4.3 Creating a Screen

Knowing the words from the last chapter we now speak the same language and can concentrate on how to design a screen (or card, if you wish) in Plus.

One of the first things I made was a flow chart of the Steritherm process. It is an extremely detailed plan and I did at that stage not know to what extent I would need special effects and action. Perhaps should it turn out that every pump and tank should be zoomable and that would require all of them to be paint objects of their own. This was made and thus it was made possible to include action into every piece of equipment on the flow chart. Because of the great number of objects on the screen if was a tedious work. That paint objects were chosen and not ordinary buttons was due to the dense packing of the objects and that irregularly shaped object would occupy less space. Another objective was to make just the object and not its surroundings sensitive to mouse clicks.

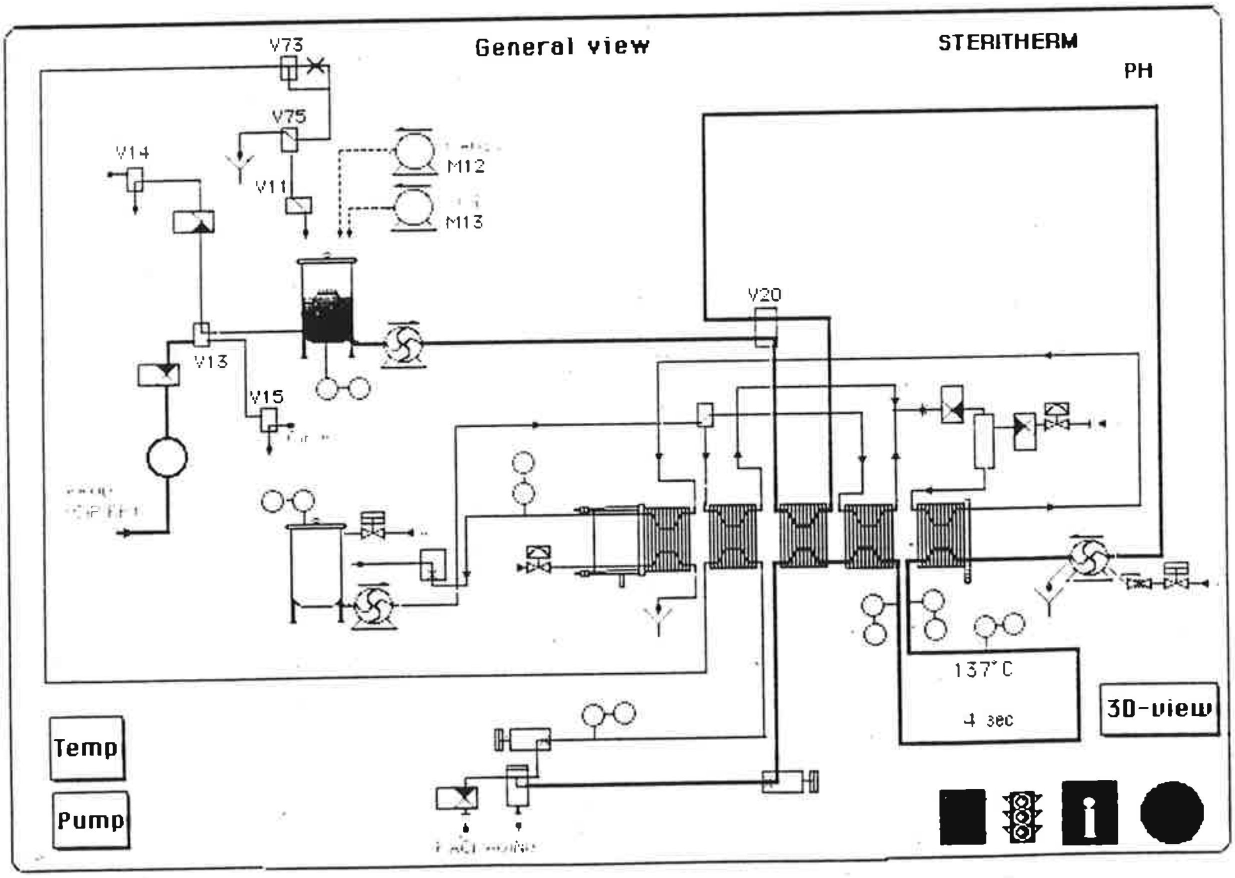


Fig 4.3.1 The Steritherm flow chart.

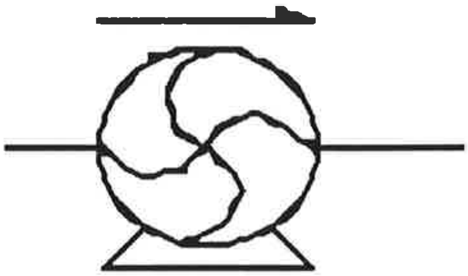


Fig 4.3.2 The flow chart symbol of a pump.

Choosing "New" from the Plus menubar a new stack was created. A double-click on the paint object icon in the toolbox supplied an area in which the first pump could be drawn. Having adjusted the size of the area, the oval tool was excellent for the creation of a circle. Anybody familiar with MacPaint knows what it is like, just hold down the mouse button and draw! Continuing with the polygon tool the pump support was finished, again quite easily.

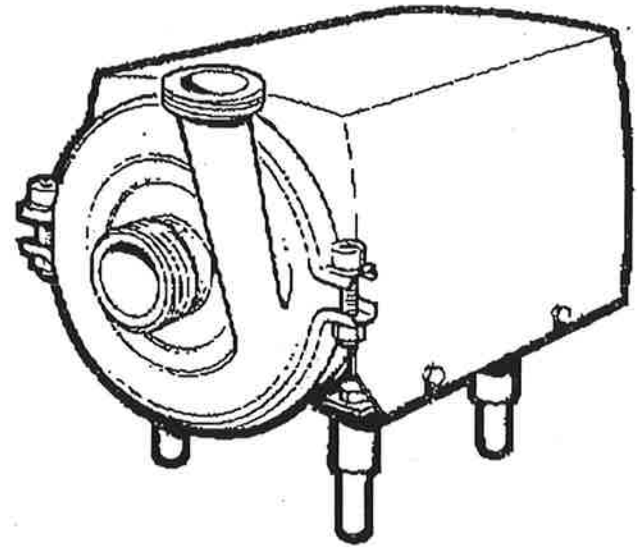


Fig 4.3.3 A scanned picture depicting the same pump as above.

The fan-like impeller blades in the middle of the pump were harder to get right. A first attempt with the pencil tool didn't succeed. The best result came with the oval tool. A small oval just fitting into half of the pump circle could be made. Dividing the oval into half and then quadruplicating one segment there suddenly were four correctly shaped impeller blades. By rotating these blades the angle could be adjusted and then they fit smoothly into the pump circle.

In much the same way the other objects on the screen were built. Having once made a pump, the work didn't need to be redone as the original was easily copied. Just like Copy and Paste in all Macintosh programs.

To improve the black and white original in Plus, it is possible to add any colour you want wherever you want it. Coloured painting could also be done directly. Both ways were fast and elegant enough that you had to be careful not to end up with a screen looking like an Impressionist painting from the late 19'th century. Tests have further proven that more than 6 colours (excluding black and white) on a screen decreases subject responses and increases errors [Wagner, 1988]. It is also noted that even fewer colours may be an advantage. A sparse and well-considered use of colour was thus executed.

A hypermedia card without action would be a waste with possibilities to help and assist the operator. Besides, it would be rather boring. An example of action is a button called "Temperature" that displays the product flow temperature in some pipes. A double-click on the button icon in the toolbox gave a fresh button. With another double-click - this time on the new button - a dialogue box opened and the desired properties like name and shape could be entered. This time a shadowed box was chosen (fig 4.3.4). A click on "Script" opened a blank sheet and the little program was entered.

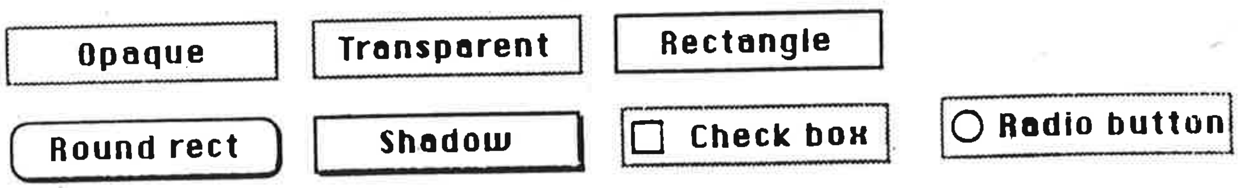


Fig 4.3.4 The different button shapes in Plus.

The idea of the temperature program is to show coloured pipes on the exact place where the black and white ones are located and then hide the latter ones. When you become tired of the coloured pipes you just click anywhere on the screen and the black ones are shown instead of the coloured. After a week this initial and in fact most complex of all screens was ready.

4.4 Creating Another Screen with a Scanned Picture

Some pictures, as the perspective "three-dimensional" view of the Steritherm, would have been too time-consuming to draw yourself. At this point a ThunderScan scanning device came in handy. It was mounted on an ImageWriter printer and did its job well, if rather slow.

On the paint object dialogue box there is an option called "Import Graphics" designed to facilitate this operation. Using it a new and clean card with only a paint object depicting a somewhat shaky and faint 3-D Steritherm was developed. It was however apparent that the quality had to be improved.

At first the contours of all parts of the picture were strengthened with the line tool. Thanks to a scaling function this wasn't too difficult as enlargements in sizes from one to eight times actually could made. Enlarged eight times it would be easy even for some one near-sighted to edit single pixels...

Having read that a grey background provides the best overall operator performance [Wagner, 1988] I experimented with one on this card. I had only to choose the paint bucket tool and the grey colour and then click on the area that should turn to grey. To be able to identify the different objects textual descriptions were inserted with the text tool. The pipes were finally coloured according to what media (water/product/condensate/steam) they carry. The attached screens in Appendix B will give you a hint of the quality of the drawings.

4.5 Linking the Screens

Each card would be of little value on its own. But linked together with other cards into what is sometimes called "an eternally cross-referenced web" new perspectives open. As a process operator you can with a clever design of buttons and paint objects easily navigate anywhere you want in the system.

The organization of the links on this interface is twofold. Firstly, you will always be able to immediately reach the four major views; the Main View, the Operator's View, the Information View and the Knowledge-Base Entrance View (fig 4.1.1). The linking is executed by a single click on one of the ever-present icons in the lower right-hand corner of the screen (fig 4.5.1). As all cards should have these icons they are put on the background.

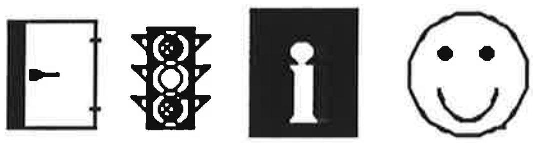


Fig 4.5.1 The icons linking you to the major views.

The second alternative to link yourself somewhere else is to click on a symbol, an icon or an object whose script contains a link instruction. Your click makes the script execute and you may for instance be zooming into an impeller inside a pump. This linking could either be done automatically or you could produce a pop-up menu with an acknowledge function. Or why not have multiple links to be able to take that close look on either the electrical, the mechanical or the photographic view. You could also include a link to the history of the pump with diagrams and charts displaying interesting process parameters. There is really no limit for the number of alternatives, except for the size of the screen.

In fact, there are two more ways to navigate through your stack. One is with the arrow keys on the keyboard and the other with a special function called "Recent". "Recent" will display miniature copies of the cards you have been visiting and by clicking on one you will be linked there.

5 A Look at the Resulting Interface

"Leibniz sought to make the form of a symbol reflect its content. "In signs", he wrote, "one sees an advantage for discovery that is greatest when they express the exact nature of a thing briefly and, as it were, picture it; then, indeed, the labour of thought is wonderfully diminished." "

-Frederick Kreiling, "Leibniz" [Schneiderman, 1987]

To strive for "the perfect screen" is a noble task but I don't think it is the most fruitful one. Being individuals we compose an enormously varied party. What one thinks is the ultimate fulfilment another may have serious doubts about. Thus I hope that the screens and views that I will present become subjected to many different opinions.

5.1 An Overview of the System

This chapter will guide you through the screens constituting a model of a process operator's interface. After a look at the physical appearance and hierarchical organization of the system the functionality will be explained.

Screens and Organization

The process operator interface is organized in what can be considered as a hierarchical structure (fig 5.1.1 and 5.1.2). This permits, among other things, a most intuitional zooming capability. By walking down into the "tree" you will be served with information and pictures that have greater detail than the ones above. Notice that you are allowed to move any way you want in the tree; even from one branch to another. Let us study the screens:

The Main View. Saying hello to the computer you will be greeted by a happy Chernoff's face telling you that everything is fine. This is the Main View and the highest level of abstraction. That is, you are as far away from the physical form of the Steritherm with motors and temperatures as you can be. You will only see the state - happy or sad - of the process by means of a Chernoff's face. From this surveying point of view you will have four options of where to go next; the Operations View, the Information View, the Economics View and the Knowledge-Base View (fig 5.1.1).

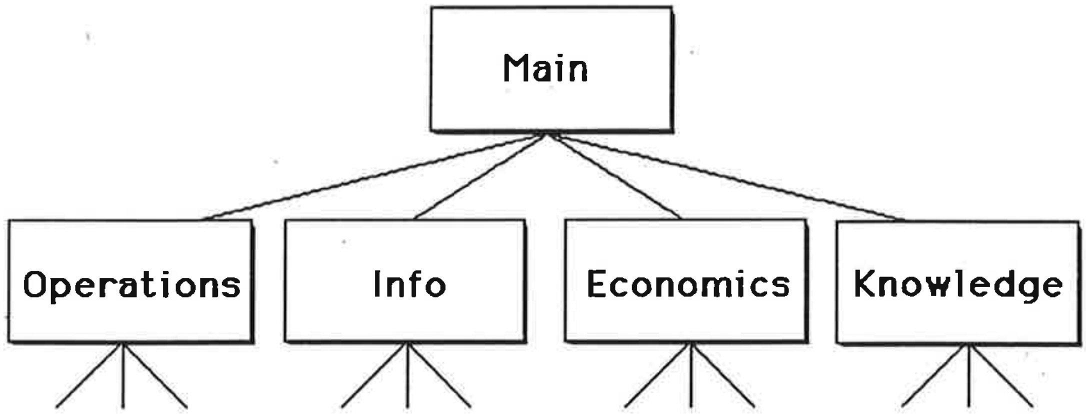


Fig 5.1.1 The top of the system structure.

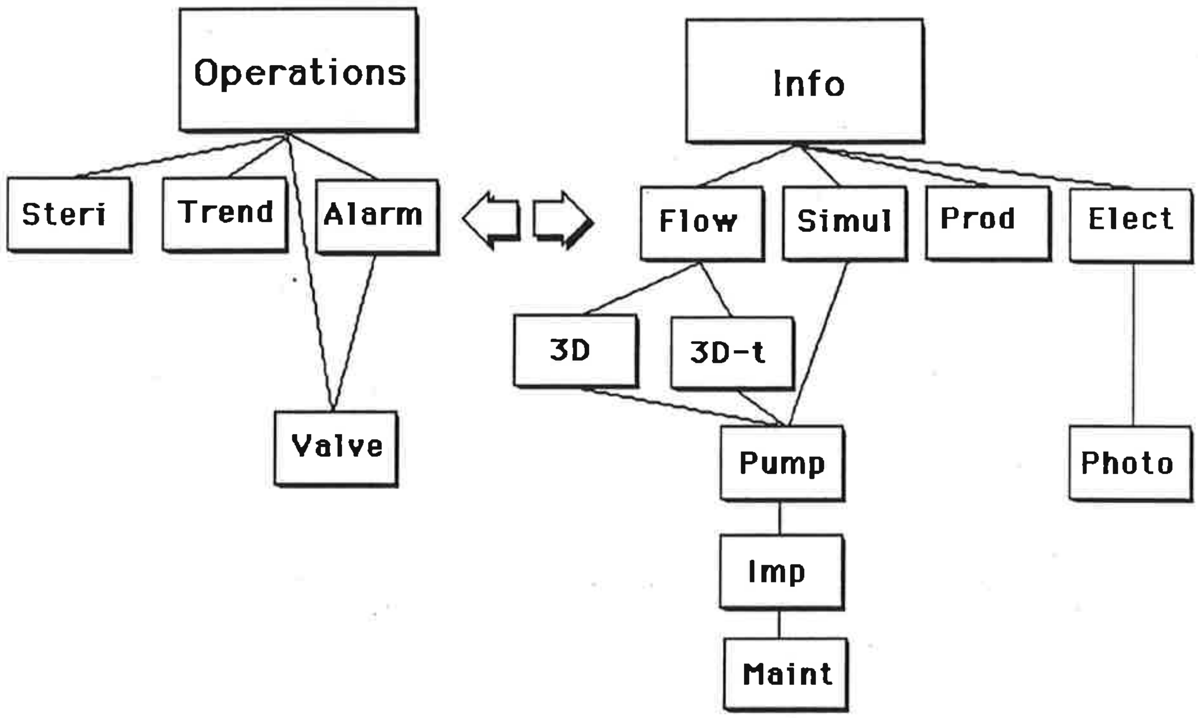


Fig 5.1.1 The hierarchical organization of two sub parts of the system.

The Operations View. In general you will go to the Operations View from which it is possible to start and stop sequences like sterilization, production, rinsing etc and check alarms. There are also several icons linking you to more information on the sequences e.g. unique flow charts and adjustment of process parameters like temperature, pressure and sterilization time. In short, you will be in control of the dynamics of the whole process from this view and its subviews.

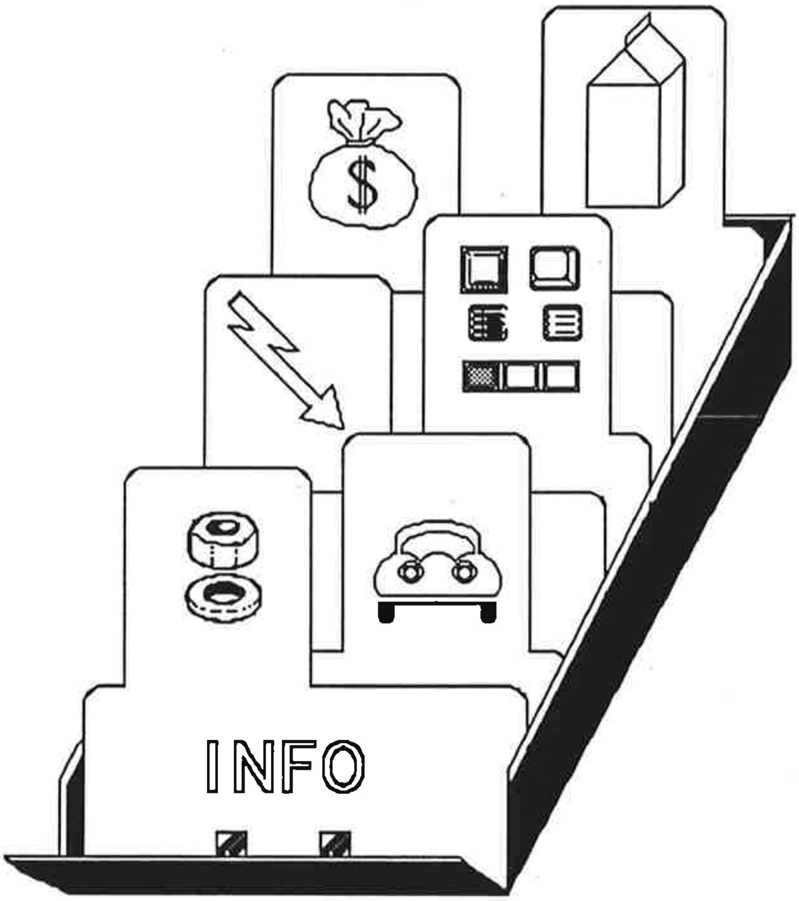


Fig 5.1.3 The Information View Chooser Box.

The Information View. The Information View gives you access to a vast sea of static information. From a register screen (fig 5.1.3) you will be able to pick what kind of information screens you are interested in and by a single click you can study for instance the complete flow chart of the process. Further on you can immediately be served with among other things colour-coded temperature indications, animated process sequences or a "three-dimensional" picture of the physical components (how pumps, tanks and other kinds of equipment are connected). All knowledge about the process will be available in this section. Multiple ways of locating information are possible and the representation will vary from simple line drawings to photographies and advanced perspective pictures (Appendix B). Colours and sounds will facilitate your search.

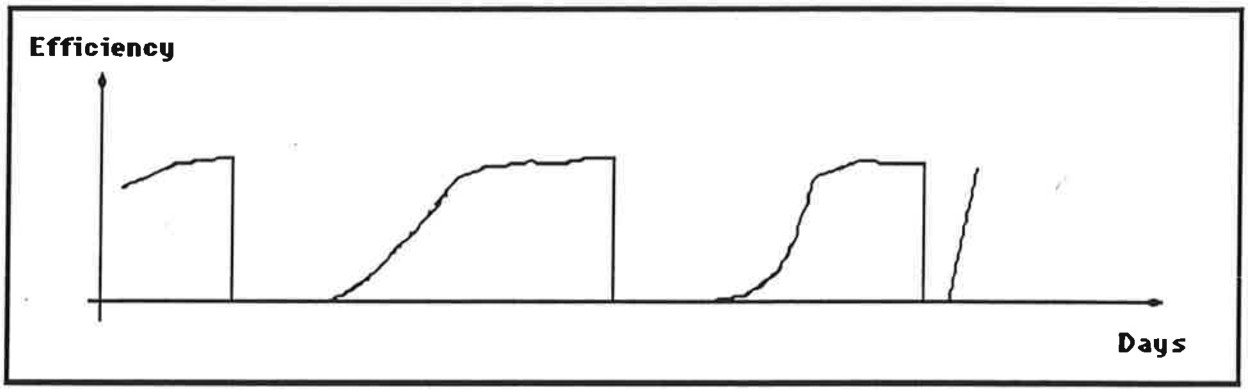


Fig 5.1.4 A process parameter recording.

The Economics View. Constituting of a mixture of dynamic (renewed all the time) and static (not subjected to change) information the Economics View will tell you how the process is being run. That is, if you are running the process with optimum efficiency. Displayed is a recording of a significant process parameter showing how the process has been run and how it is presently being run (fig 5.1.4). This option is commonly found in modern process control systems.

The Knowledge-Base View. The fourth part is a link to the common knowledge-base itself. The thought is (meaning that I haven't implemented it) that from this screen you will have access to the common knowledge-base containing the underlying process and control knowledge. The knowledge will be represented in terms of hierarchical objects, control blocks, rules, simulation equations, functional and qualitative descriptions etc. This part of the system is constructed by Pär Ericsson and Mikael Christiansson as their Master Thesis. They use the G2 system on a Symbolics Lisp Machine. Figure 5.1.5 will show you their flow chart which - as intended - is quite similar to my flow chart.

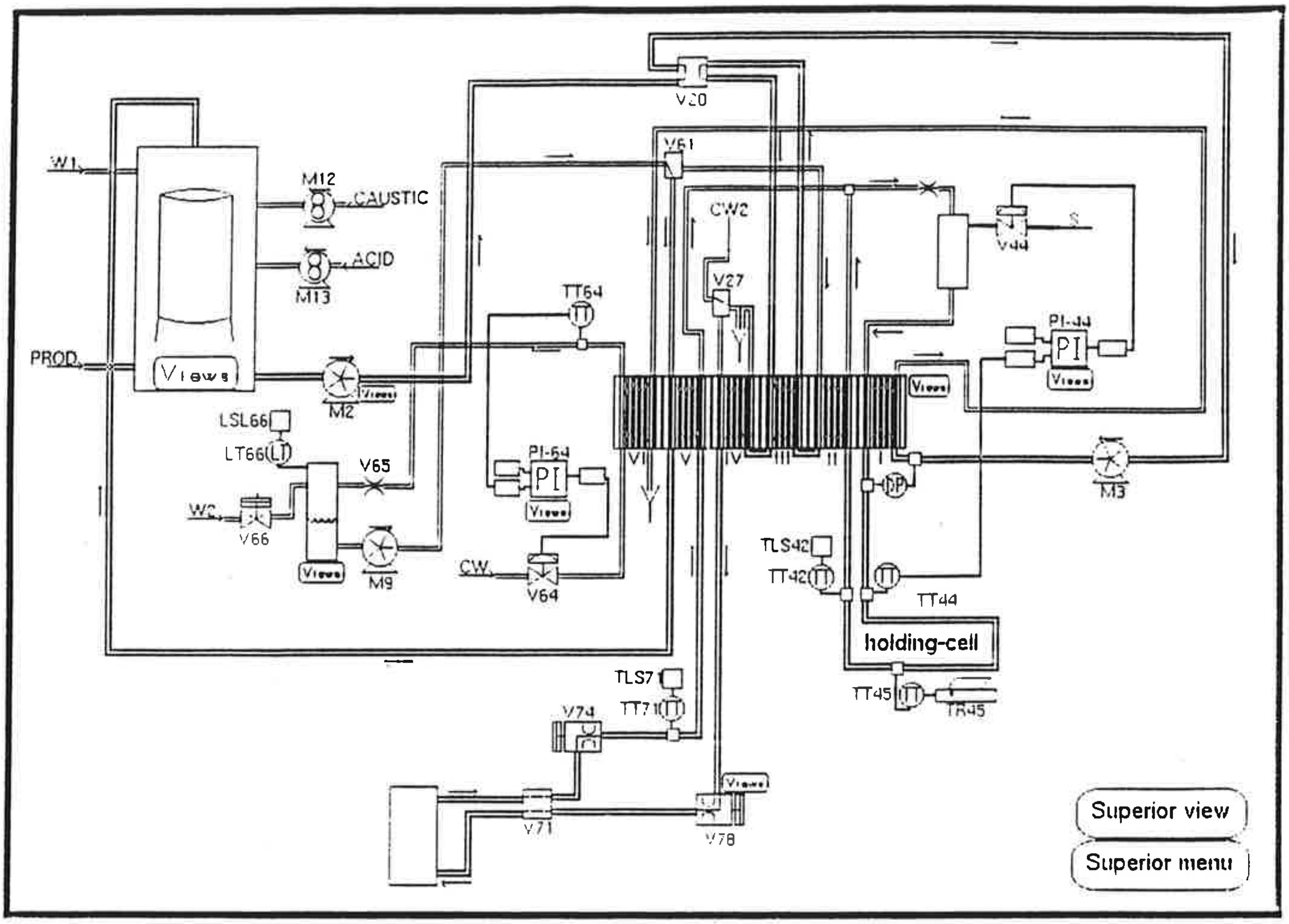


Fig 5.1.5 The Steritherm flow chart in the G2 Master Thesis.

Functionality

As has been explained in previous chapters each and every action can be initiated with a single mouse click. The type of action may vary from merely a change of screens to the start-up of a new process sequence. All screens have some functions and a predestinated alarm area in common. In the hypermedia programs you say that these items are positioned on the background.

The alarm area is located at the very top of each screen. Should an alarm be raised no matter where you are in the system this area will flash with a message specific for both type and location of the alarm. It could e.g. be "Motor alarm M6" (fig 5.1.6). After an initial deep female sigh from the computer (sounding "Oh, shit!") a highly annoying acoustic alarm signal will begin to sound. The signal won't end until the operator has acknowledged the alarm by clicking with the mouse. To remove the alarm text yet another click on an acknowledge button is demanded.

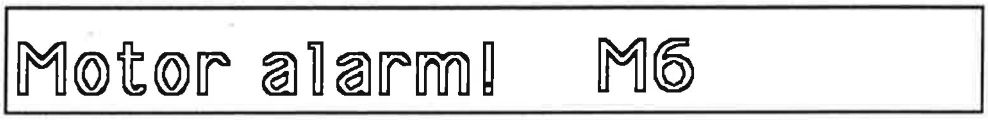


Fig 5.1.6 The alarm area and a message.

When an alarm has been activated an icon saying "WHY??" will be displayed (fig 5.1.7). A click here will give the operator access to a diagnostic expert system to locate and cancel the cause of the alarm. As such an expert system hasn't been incorporated into my model the operator will be given a message that no help can be offered. It is fact the HAL computer from the science fiction movie "2001" saying "-I'm sorry Dave, I'm afraid I can't do that". It is with this feature like so many others not the actual choice of wording or symbol that is important but the clear demonstration that these things let themselves be done! In this case more appropriate (but perhaps less fun) messages can easily be recorded using a MacRecorder audio digitizing set.

Fig 5.1.7 The icon giving you access to a diagnostic expert system.

Once an alarm has been activated a red alarm light will be lit on the Operations screen indicating that an alarm is present. By clicking on it you will be linked to an alarm list recording more facts about the alarm. A possible extension of this system would be to let further clicks on the displayed text link you to yet more detailed information. Plus gives in its programming language support for an accomplishment of such an extension.

In this prototype no alarms are raised by themselves - although that too could be programmed. Instead, they are the result of a simulation. You will find the simulation button (fig 5.1.8) on the Main screen which will display a red-coloured and sad Chernoff's face once the alarm has been triggered. Not until the cause of the alarm has been removed with a click on the WHY??-icon the happy and green Chernoff's face will reappear.

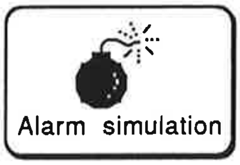


Fig 5.1.8 The alarm simulation button.

A leading thought through-out the system is that the operator shouldn't be confronted with more information than is manageable. He shall however at any time be able to get the most detailed information available on all parts of the system. As an example of the advantages of this hierarchical organization I will mention the zooming possibility of one of the product pumps.

Using the click technique on the pump either on the flow chart or the "3D-view" (both described in chapter 4.3) you will be taken to a screen showing a scanned picture of the pump. There are also icons, giving you access to characteristic pump charts, and a scrolling field with lots of information on the pump.

You can now browse through the information to find the data you were looking for or choose to zoom further. A click on the pump impeller will show you another screen with an impeller close-up (fig 5.1.9) and additional information . A click on the impeller axis will, accompanied with a "zoom-in-sound", move you to an extremely detailed zoom picture of the smallest components down to the last screw and washer. Any component can be click on and will then display its name, serial number, price and the number in store. I will have to admit that this last screen is a slightly modified demo screen delivered with Plus on purchase, but at the same time it is a good illustration of the possibilities to use "old" material.

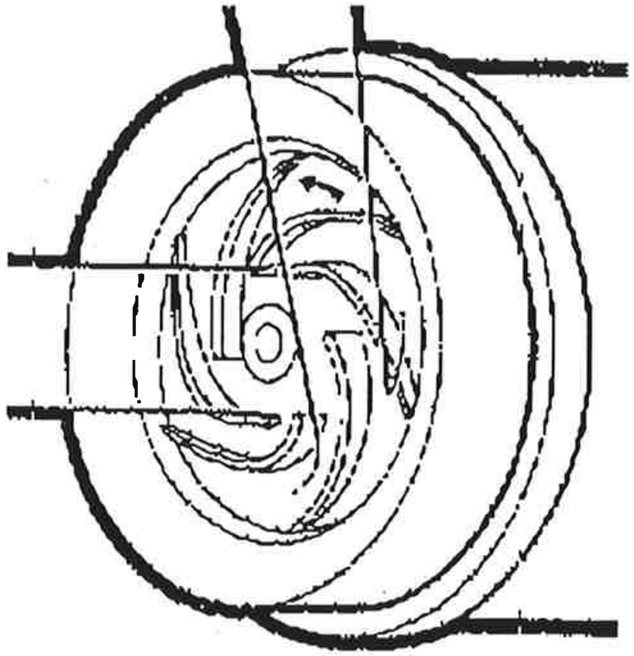


Fig 5.1.9 An impeller close-up (scanned).

Any time you want you can go to the previous view or to a view with more detail. You can also choose to go to any of the four major screens by clicking on the ever-present icons in the lower right-hand corner. This will help you never to get lost in the system and at the same time allow you great freedom of choice.

In much the same way it will be possible to control the process from different levels of detail. Ordinary start/stop procedures can be done from the Operations View, which basically is rather similar to the operator's control panel of the Steritherm today. Adjustments of process parameters must however be done on a deeper level like the individual sequence screens. If you want you can change from the dynamic view-point with its parameter adjustments to the static view-point with information zooming. Just as always it is done with simple mouse clicks.

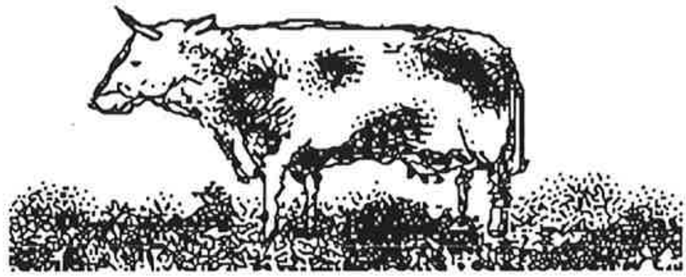


Fig 5.1.10 A typical Swedish cow.

The final views I would like to tell you about are reached from the Information View. One branch will take you to the electrical information of the Steritherm providing electrical schemes, circuit specifications, photographs etc. The other branch consists of only one screen with info on the product. As we are dealing with the sterilization of milk in this model the screen is decorated with a hand-painted picture of a cow (fig 5.1.10).

5.2 Connecting to the Knowledge-Base

In my prototype no real knowledge-base exists and neither does the expert system that would have been required to fetch the data from it. Instead knowledge has explicitly been inserted into the program. This is of course not intended in a real world implementation. Then you would probably want to store all the knowledge about the process in a common knowledge-base forming the heart of the system.

It will be necessary to impose an object-oriented nature to the knowledge-base as object-orientation is a suitable way to achieve good interacting models of the process components and the control system. Further on, the knowledge-base would probably be organized in a hierarchical way to conveniently be able to handle large amounts of information. To assist the user you would need an expert system and thus you will have to include modes and rules into the knowledge-base. There will of course also be a real-time link to the process itself.

This Master Thesis has resulted in a multimedia model emulating the operator interface in such a system.

5.3 Improvements upon this System

Let me first state that this is merely a first prototype and therefore many improvements can be made. A great number of features have been exemplified in depth but on a few objects only. Had it been done on more objects e.g. interconnecting links between close-ups could have been useful to an operator.

Another feature that would be both desirable and achievable would be high-lighting of chosen objects i.e. when you come back from a zoom to a flow chart the object you had studied should be illuminated. The same kind of illumination could be applied to an object that is the subject of an alarm.

As an improvement regarding the alarms, the possibility to let the operator zoom into more detail from the alarm list has already been mentioned. In an emergency situation an operator could also benefit from detailed written procedures of how to handle the situation [Gertman et al, 1985]. These could for example be available through the not implemented expert system or from an icon of its own. This icon ought to be located in such a way that it could easily be reached during an alarm but also be reachable at other times. A suggestion is to put one icon on the alarm area beside the WHY??-button and another one the alarm list screen where it always would be present.

To make the system complete the operator should be able to study process parameters by clicking on a transmitter. He could then, if desired, be provided with a drawing of the transmitter as well as a photograph along with characteristic information. He could also choose to see trend curves ("-How has the temperature has been changing the last few hours?") presented on easily read displays and graphs.

These are just some of the possible improvements that I can come up with. Together with the projected user of the system many other features could be invented and added. Already implemented ones would also receive proper modifications.

6 Conclusions

"What needs to be articulated, regardless of the format of the man-machine relationship, is the goal of humanism through machines."

- Nicholas Negroponte [Brand, 1987]

One of the objectives of this report has been to document the design of a process operator interface in future real-time knowledge-based control systems. Many recent ideas from present man-machine communication research have been incorporated into the system. The objective was to find out if these theoretical ideas could be implemented in a smooth way so that you could get a better feeling of their meaning and suitability.

Hypermedia programs were chosen for the above purpose and to try out their applicability as a prototyping environment was a third and major objective. I will in the following chapter conclude my experiences in these matters and finally present some ideas for the future.

6.1 Conclusions About the Interface

The main purpose of the interface is to make it easy for the process operator to rapidly extract valid information from the system and respond to it.

A complex system receives and processes thousands of signals each second, which to a human being would be an overwhelming amount of information. To cope with this problem different levels of abstraction have been introduced. A strict hierarchical organization of facts that allows for information zooming by simple mouse clicks diminishes the problem further.

The integration of textual information with video films, animations, drawings, flow charts, parameter diagrams, simulations and other presentation forms enables a high quality of the man-machine communication. Reduced time to master the system can be foreseen as meters of instruction books and manuals are incorporated into the computer system. This provides fast and easy access along with possibilities for optimum choice of presentation techniques (to see a short film may be more instructive than to read a chapter in an instruction book).

The hierarchical organization of the information facilitates rapid decision-making. Fast operator responses are further supported by a clear and consistent use of colours and icons along with the choice of the mouse as the single interaction device.

6.2 Conclusions About Hypermedia and Plus

Traditional models of software development assumed that designers could stabilize and freeze system requirements. This isn't the case in practice as users first need some experiences with the software system before they know exactly what they want. Rapid prototyping solves this problem and it thus supports software evolution and initial development [Luqi, 1989]. It is my opinion that hypermedia is a well suited environment for this kind of rapid prototyping.

Personally, I have experienced hypermedia as an easy and fast way to build any kind of small system. My judgement is however only based on literature studies and some months work with HyperCard and Plus. These programs are far from the only representatives of the hypermedia concept. Some purists even think that they aren't true hypermedia programs as they lack certain features like multiple window displaying. The small systems mentioned may for example be an address register with automatic phoning and form printing, an emergency system for environmental accidents, or an operator's interface.

The capabilities of the hypermedia program Plus have however been essential to realize the interface performance mentioned in chapter 6.1. Easy linking, good graphics, inclusion of scanned pictures and integration of exterior devices have been very valuable. Plus' support for colour and sounds have been most appreciated.

The drawbacks of Plus are primarily slow speed and a lack of multiple window / parallel processing capabilities. A better manual and help-function would have reduced learning and development time. However, anybody familiar with HyperCard, which is indeed easy to learn, will have no difficulties to learn Plus and will soon be able to rapidly develop prototypes.

6.3 About Where To Go Next

Extending our look a little longer into the future and beyond the next three to five years we see a new generation of human-computer interaction devices coming up. We will probably have computers that can read our lips and eyes, which can feel like they are reading our minds [Brand, 1987]. A lot of effort is being put into the research on voice recognition and eye-tracking and prototypes are already tested at MIT in Boston.

Research is also being made on haptic modality - touch, pressure, temperature, etc - the info we get by handling things. As these techniques evolve, continuous testing must be performed to find out the usefulness and applicability of these visions. Inspiring reading on this subject will be found in "The Media Lab" by Steward Brand [Brand, 1987] and of course in the science fiction literature.

Fig 6.3.1 A satisfied Macintosh user.

6.4 About How To Get a Demo of the Result

The emphasis of this Master Thesis has been put into the development of the process operator's interface and the exploration of hypermedia as a prototyping environment. In my work I have used colour and on-screen-action as much as possible, which is extremely hard to describe properly in a short report. If you have any questions regarding this Master Thesis or if you would be interested in a demonstration of the process operator's interface please don't hesitate to contact me. I can be reached through:

Department of Automatic Control
Lund Institute of Technology
att: Karl-Erik Årzén/Peter Höjerback
Box 118
S-221 00 Lund
SWEDEN

References

ABB AB, SattControl AB, TeleLOGIC AB, & Department of Automatic Control at LTH (1988).
Knowledge-Based Real-Time Control Systems, IT 4 Feasibility Study.
Lund: Studentlitteratur.

Alfa-Laval AB.
Dairy Handbook.
Lund: Alfa-Laval Food Engineering.

Alfa-Laval AB.
UHT Steritherm Instruktionsbok.
Lund: REF no:N 172580.

Alfa-Laval AB.
UHT Elektriska Instruktioner.
Lund: REF no:N 172580.

Apple Computer, Inc. (1987).
Macintosh Systemprogram Handbok Version 6.0.
Cupertino, California: Apple Computer, Inc.

Apple Computer, Inc. (1987).
HyperCard Användarhandbok.
Cupertino, California: Apple Computer, Inc.

Apple Computer, Inc. (1987).
Macintosh Systemverktyg Handbok.
Cupertino, California: Apple Computer, Inc.

Apple Computer, Inc. (1987).
AppleShare User's Guide.
Cupertino, California: Apple Computer, Inc.

Apple Computer, Inc. (1986).
LaserWriter and LaserWriter Plus.
Cupertino, California: Apple Computer, Inc.

Apple Computer, Inc. (1986).
Macintosh II.
Cupertino, California: Apple Computer, Inc.

Brand, Steward (1987).
The Media Lab. Inventing the Future at MIT.
New York: Viking Penguin.

Claris Corporation (1987).
MacDraw II.
Claris, California : Claris Corp.

Dhaliwal, D.S. (1985).
The Use of Artificial Intelligence in Maintaining and Operating Complex Engineering Systems.

Ehn, P., & Kyng, M. (1984).
A tool perspective on design of interactive computer support for skilled workers.
Unpublished manuscript, Swedish Center for Working Life, Stockholm.

Format Software GmbH (1989).
Inside PLUS.
West Germany : Format Software GmbH.

Gertman, D.I., Haney, L.N., Jenkins, J.P., & Blackman, H.S. (1985).
Operator Decision Making Under Stress.
Conference proceedings, IFAC Man-Machine Systems, Varese, Italy.

Goodman, Danny (1987).
The Complete HyperCard Handbook.
New York: Bantam Books.

Goodman, Danny (1989).
HyperCard-handboken.
Lund: Studentlitteratur.

Hoggard, Nick (1989).
Preliminary Specification of Multimedia Model of Steritherm Process.
Unpublished memo: ABB Corporate Research / KLL.

Johannesson, Göran (1989).
Arbeta effektivt med Macintosh.
Lund: Studentlitteratur.

Luqi, (1989).
Software Evolution Through Rapid Prototyping.
Computer (IEEE), May 1989, pp 13-25.

Nelson, T.H. (1977).
The Home Computer Revolution.

Norman, Donald A., & Draper, Stephen W. (1986).
User Cantered System Design.
Hillsdale, N.J.: Lawrence Erlbaum Associates.

Operation of the OPC5 Control Panel (1988).
Unpublished operation manual, Isopac, Belgium.

Rosenberg, B., Rytoft, C., Åberg, A., & Årzén, K.-E. (1989).
Evaluation of the Sterilab-Steritherm Process.
IT 4 report: ABB AB, SattControl AB, Dept of Automatic Control at LTH.

Scneiderman, Ben (1987).
Designing the User Interface.
Massachusetts: Addison-Wesley Publishing Company.

Thunderware (1984-1987).
ThunderScan User's Guide.
California: Thunderware.

Wagner, Eric (1988).
The Computer Display Designer's Handbook.
Lund: Studentlitteratur.

Årzén, Karl-Erik (1989).
Steritherm - Views and Hierarchical Levels.
Unpublished memo: Dept of Automatic Control at LTH.

Årzén, Karl-Erik (1989).
Knowledge-Based Control Systems. A G2 Prototype.
Unpublished memo: Dept of Automatic Control at LTH.

Appendix A: Process Definition

The following text is from the "IT4 Feasibility Study" named "Knowledge-Based Real-Time Control Systems" [ABB AB et al, 1988].

"The topic of this project is real-time, knowledge-based process control. Process control is the sum of the different tasks that interact with a specified process and with the different users of the process. In the context of this paper a process is the controlled flow of matter, energy, or information (sink) (DIN, 1985). The flow may be continuous or discontinuous. The process control system is the system that controls and supervises the operation of the process.

This definition of process includes the process industry, the manufacturing industry, and the telecommunication systems. As pointed out by Dhaliwal (1985), processes in these domains have a number of common features:

-The complexity of the systems is such that no single individual or small group of individuals can fully understand them.

-The operations and maintenance manuals may cover several tens of volumes. Maintenance of the documentation is a particular problem. It is difficult to access relevant and correct information speedily.

-Systems are continually changing and evolving since: the process and the operating environment changes; shortcomings in the original specifications come to light; bugs are discovered; and, technology advances.

-The rate of change means that old methods of training and retaining staff no longer are adequate.

-Different users of the systems need markedly different styles of interaction with the system.

-Speedy and accurate correction of faults is required. The hazards of slow or incorrect treatment are:

Faults not treated early enough may propagate catastrophically.

Appendix A: Process Definition (cont'd)

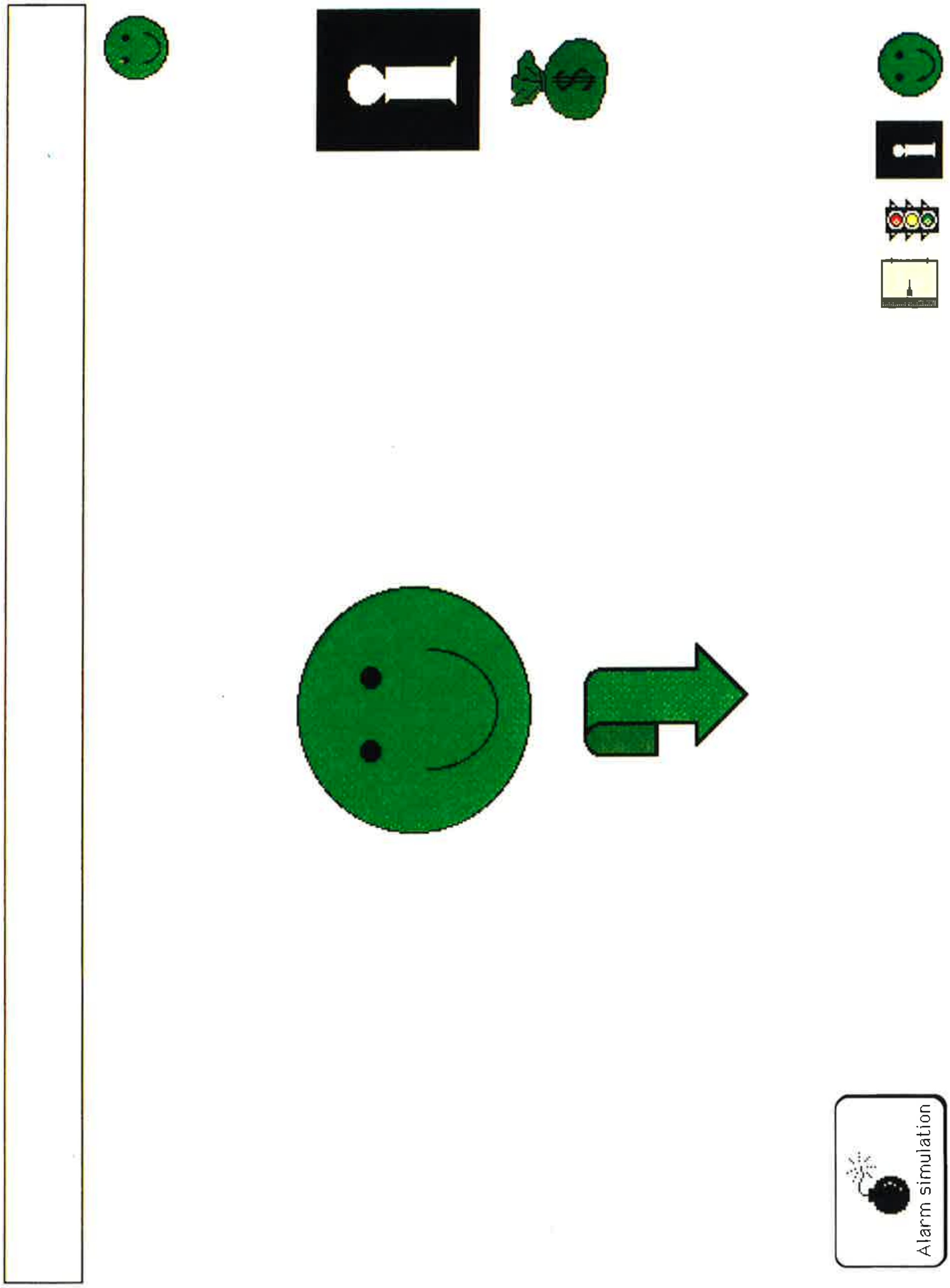
Dormant faults undetected or left untreated may greatly affect the overall reliability and maintainability of the system.

The existing control systems may itself be a prone to failure and thus may mask the true cause of misoperation.

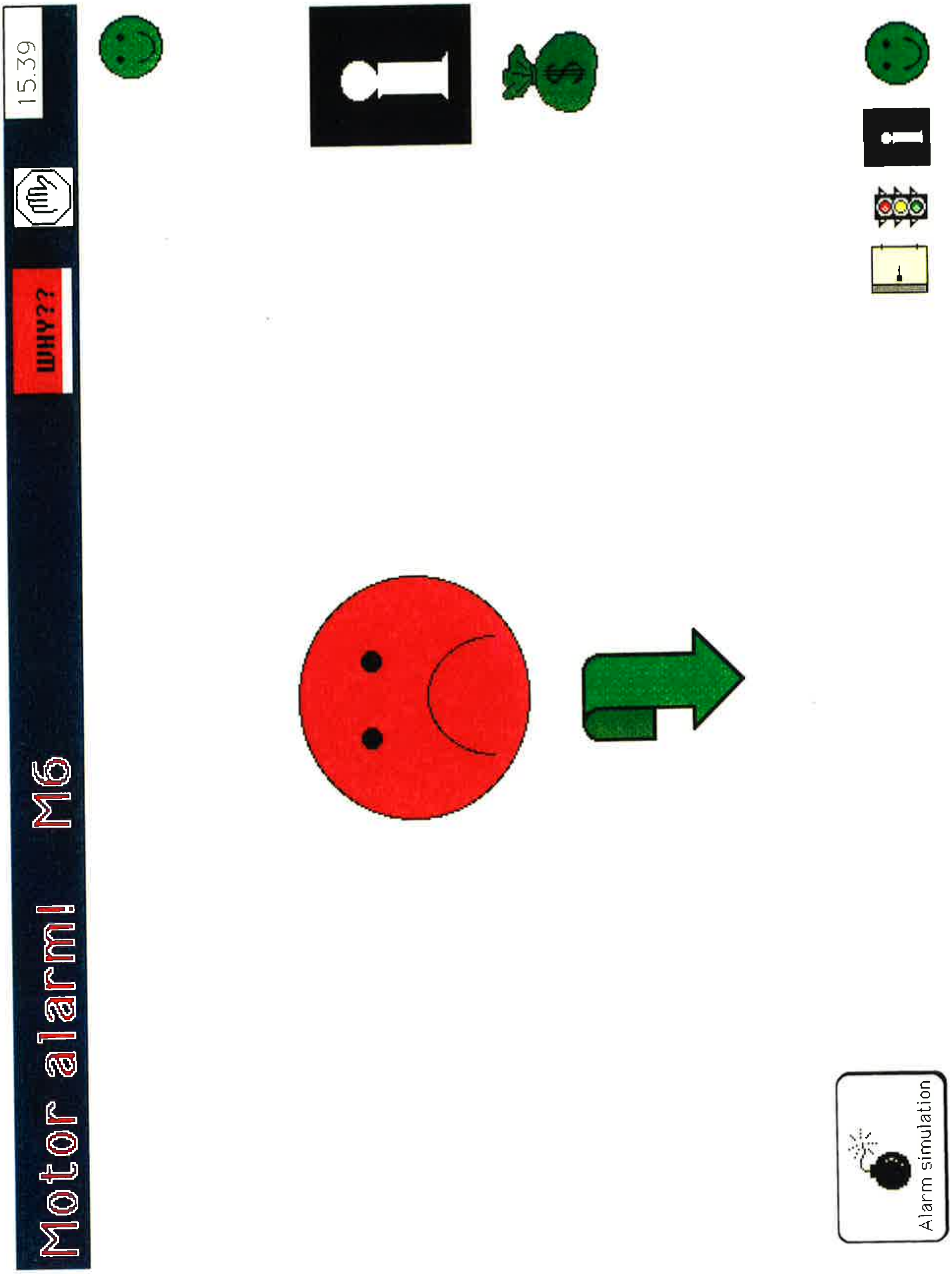
Wrongly identified faults and consequential repair actions may make matters worse.

The high reliability of the systems gives problems. Some failures are so rare that it is difficult to ensure that maintenance staff are appropriately predisposed or equipped to handle them."

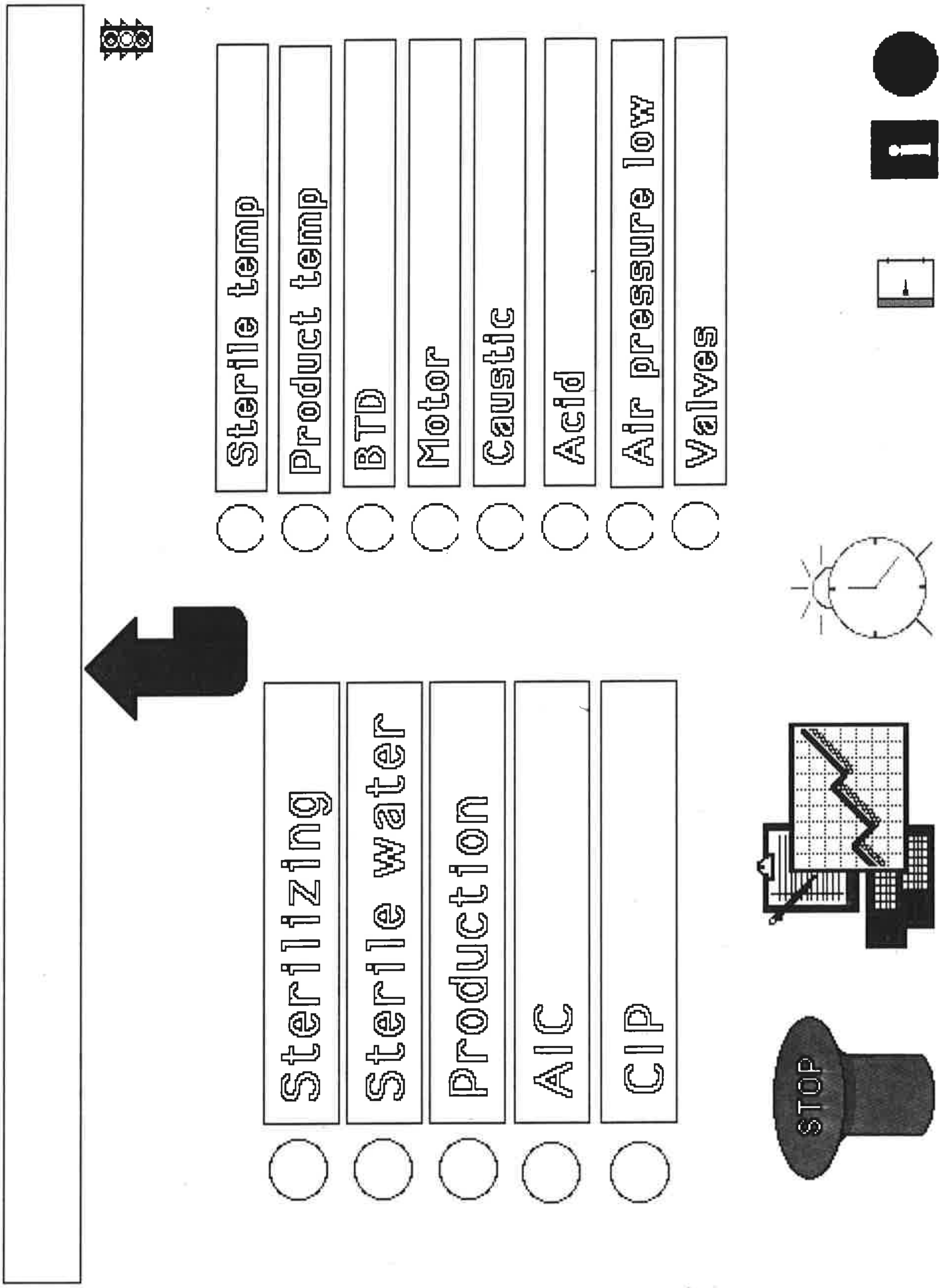
Appendix B: The Main View - Normal



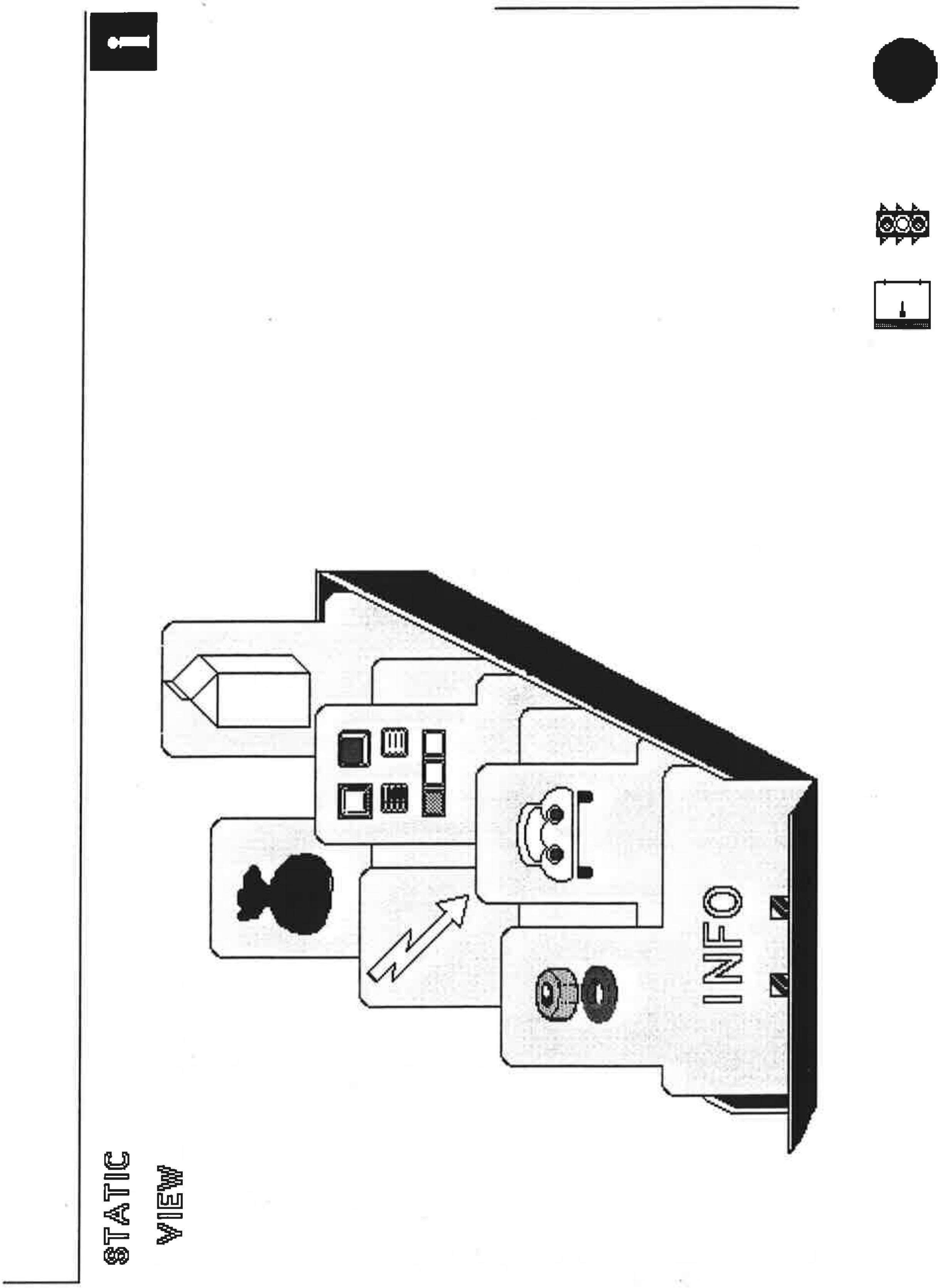
App B: The Main View - Alarming



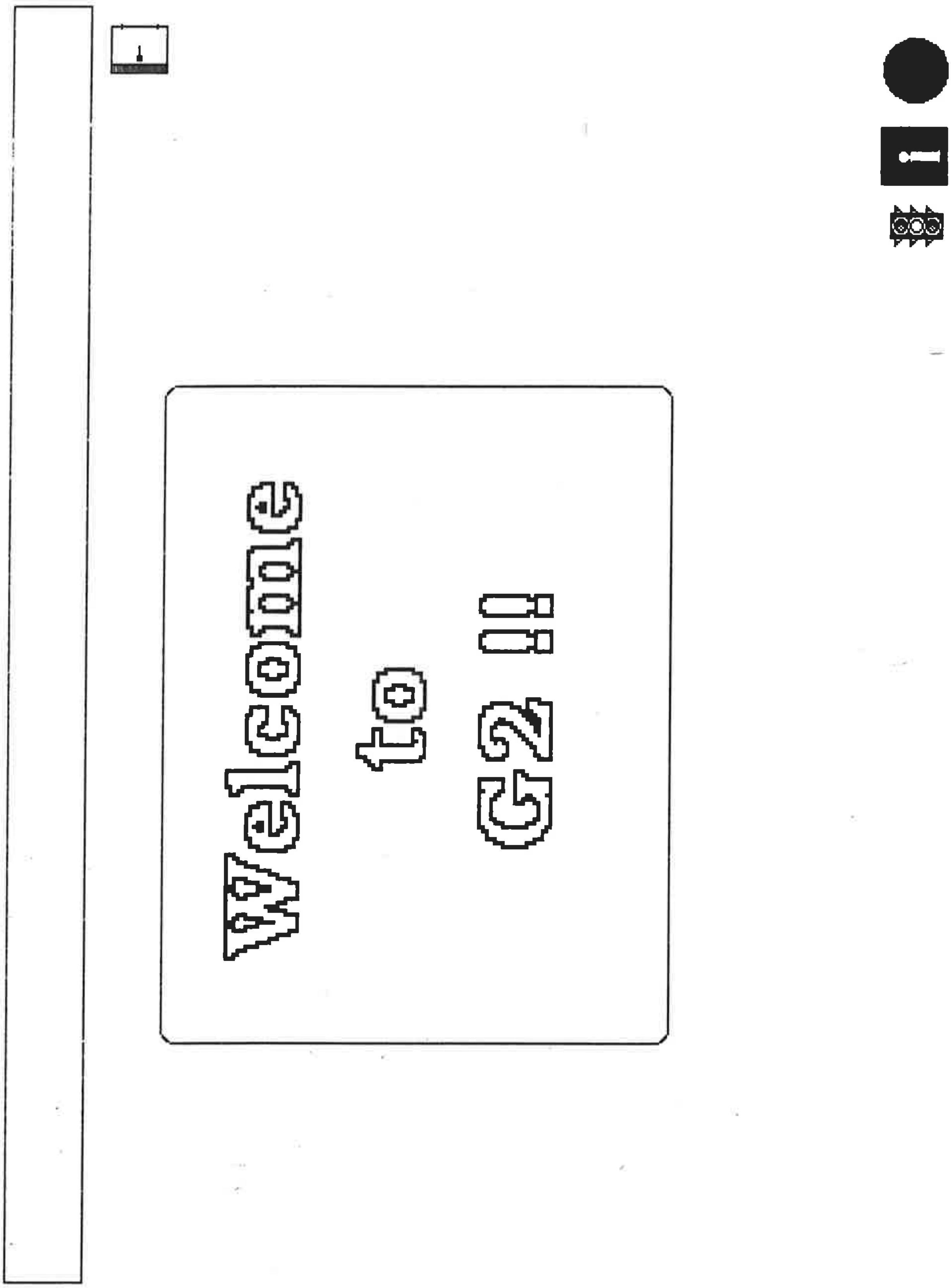
App B: The Operations View



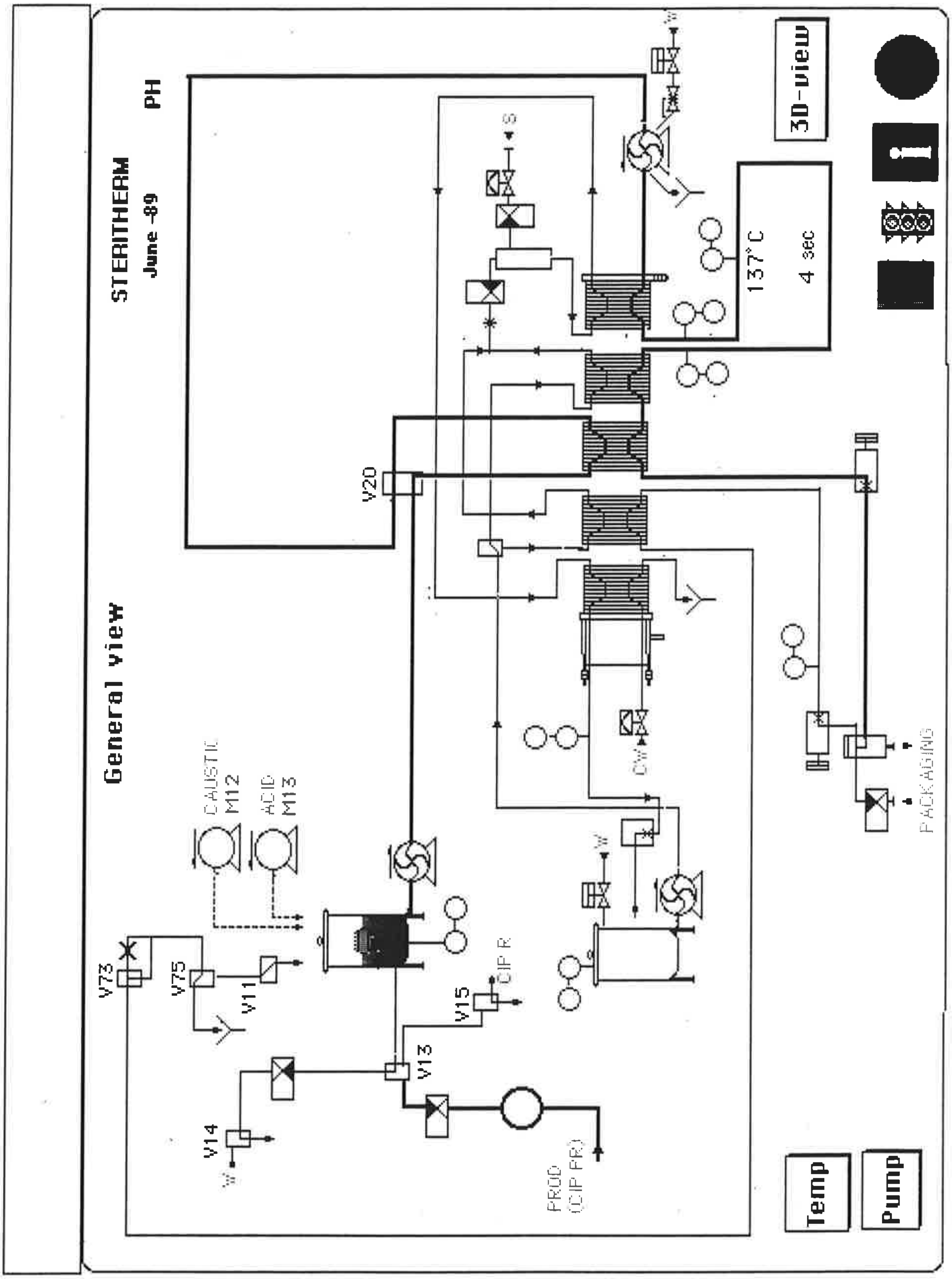
App B: The Information View



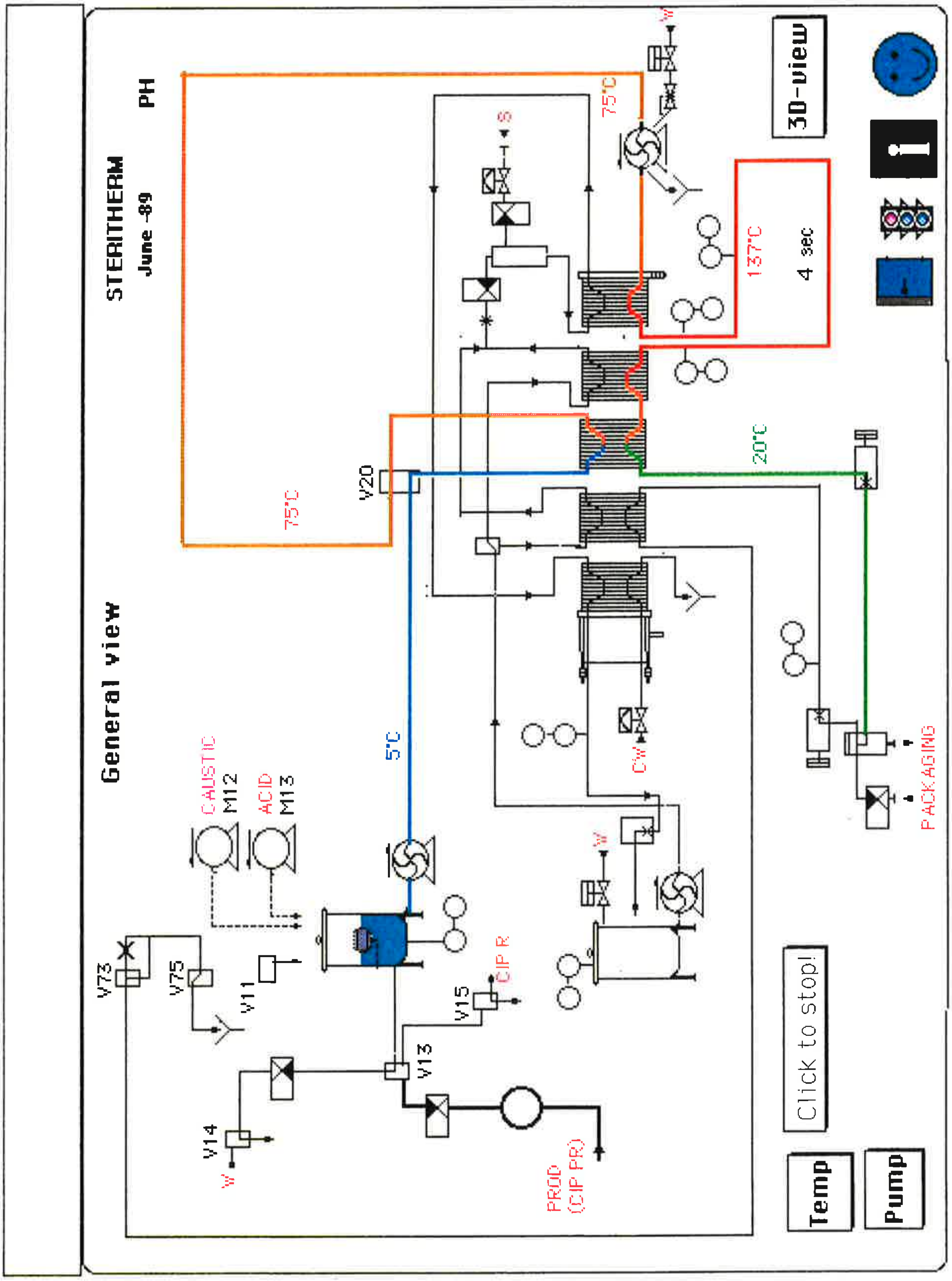
App B: The Knowledge-Base View



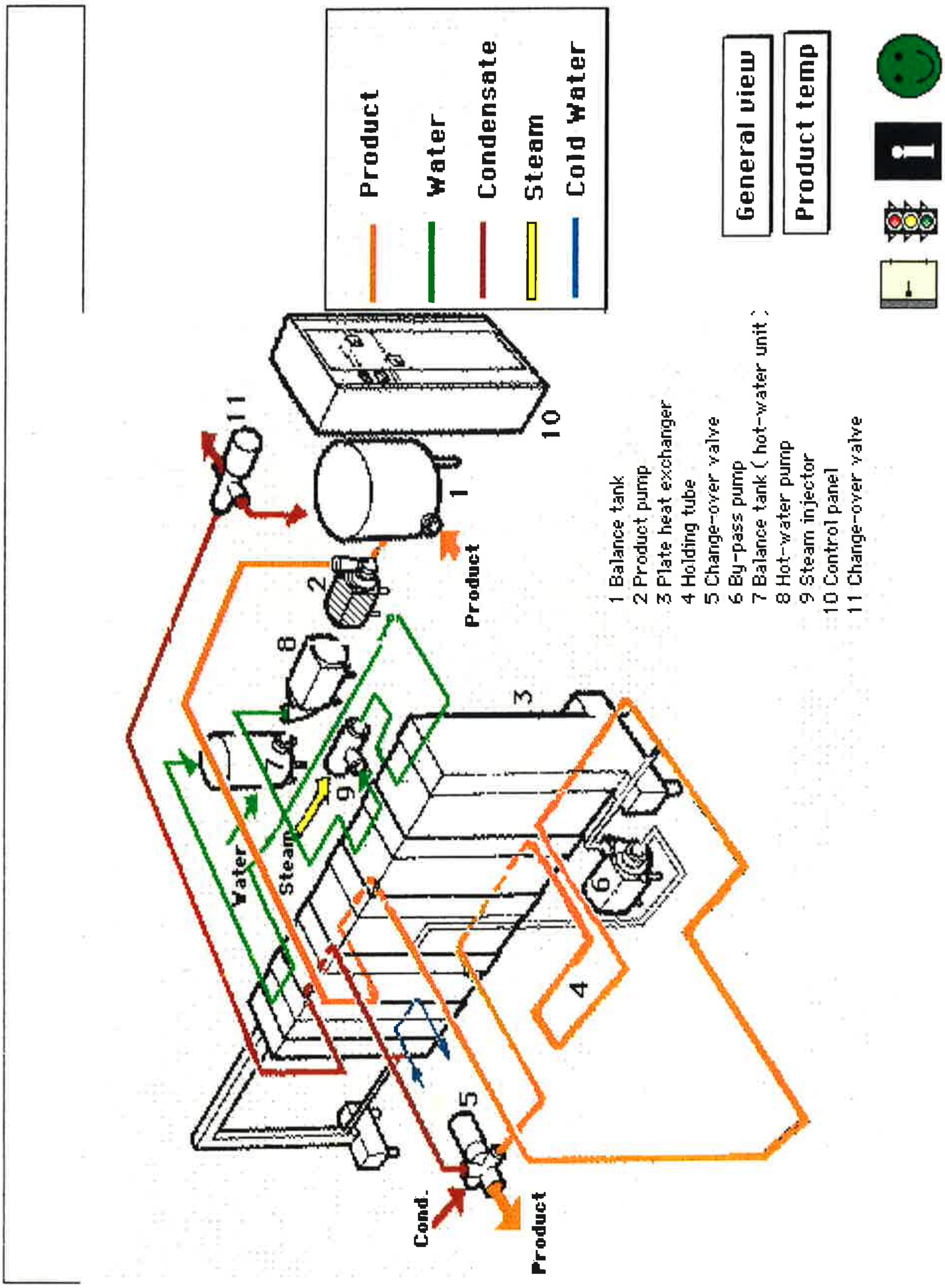
App B: The Flow Chart View - Normal



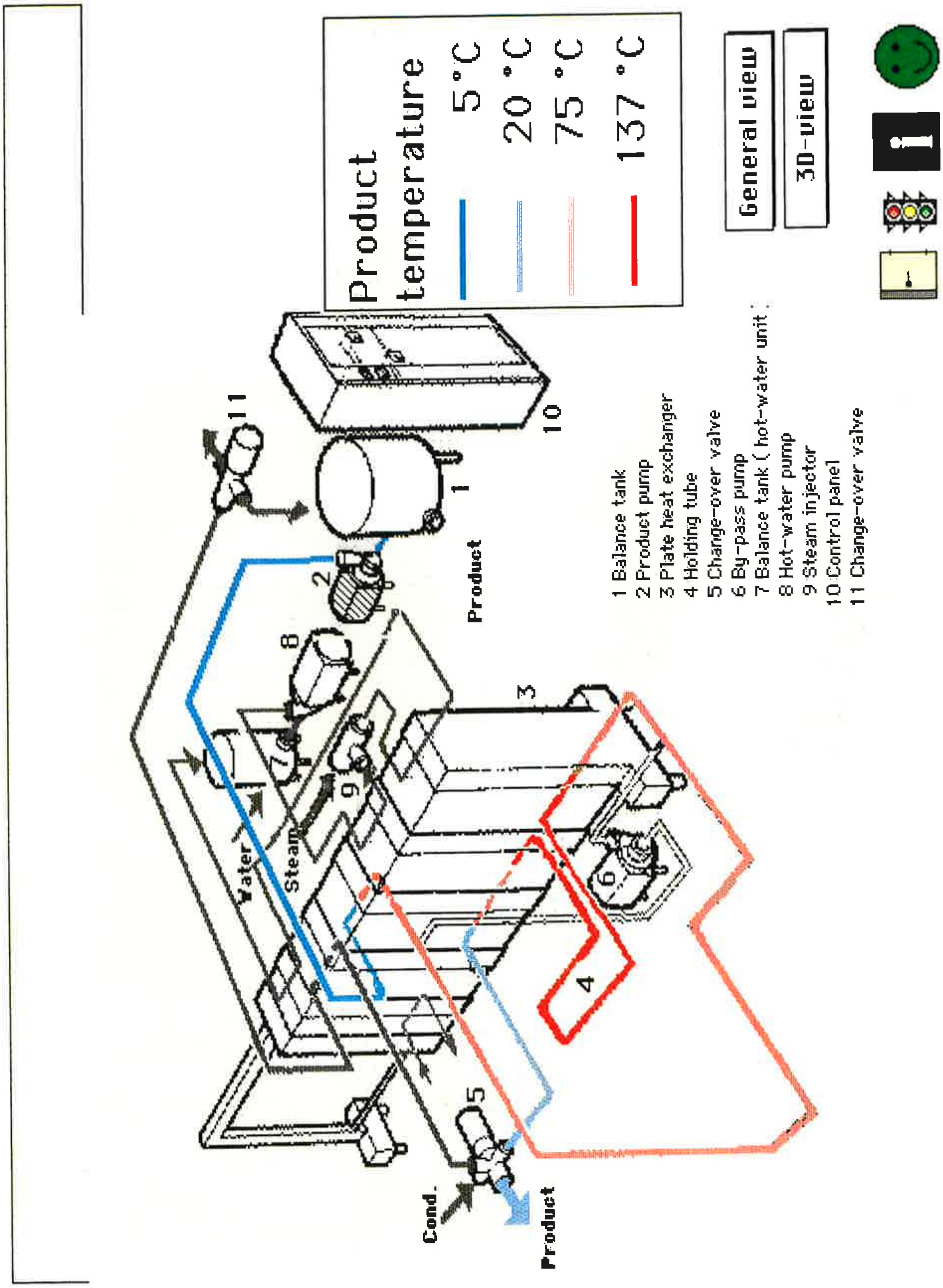
App B: The Flow Chart View - Temperatur



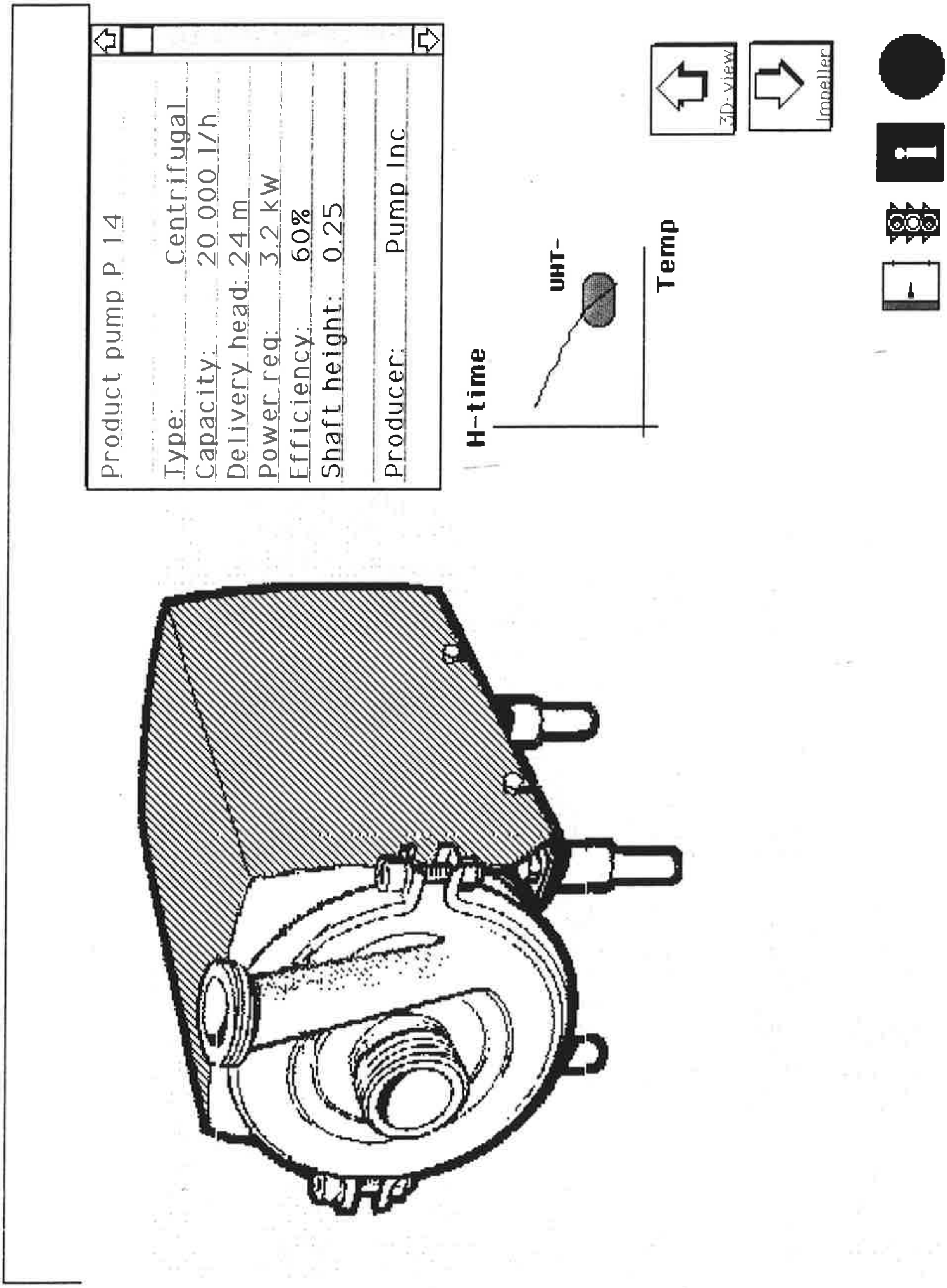
App B: The 3D View - Normal



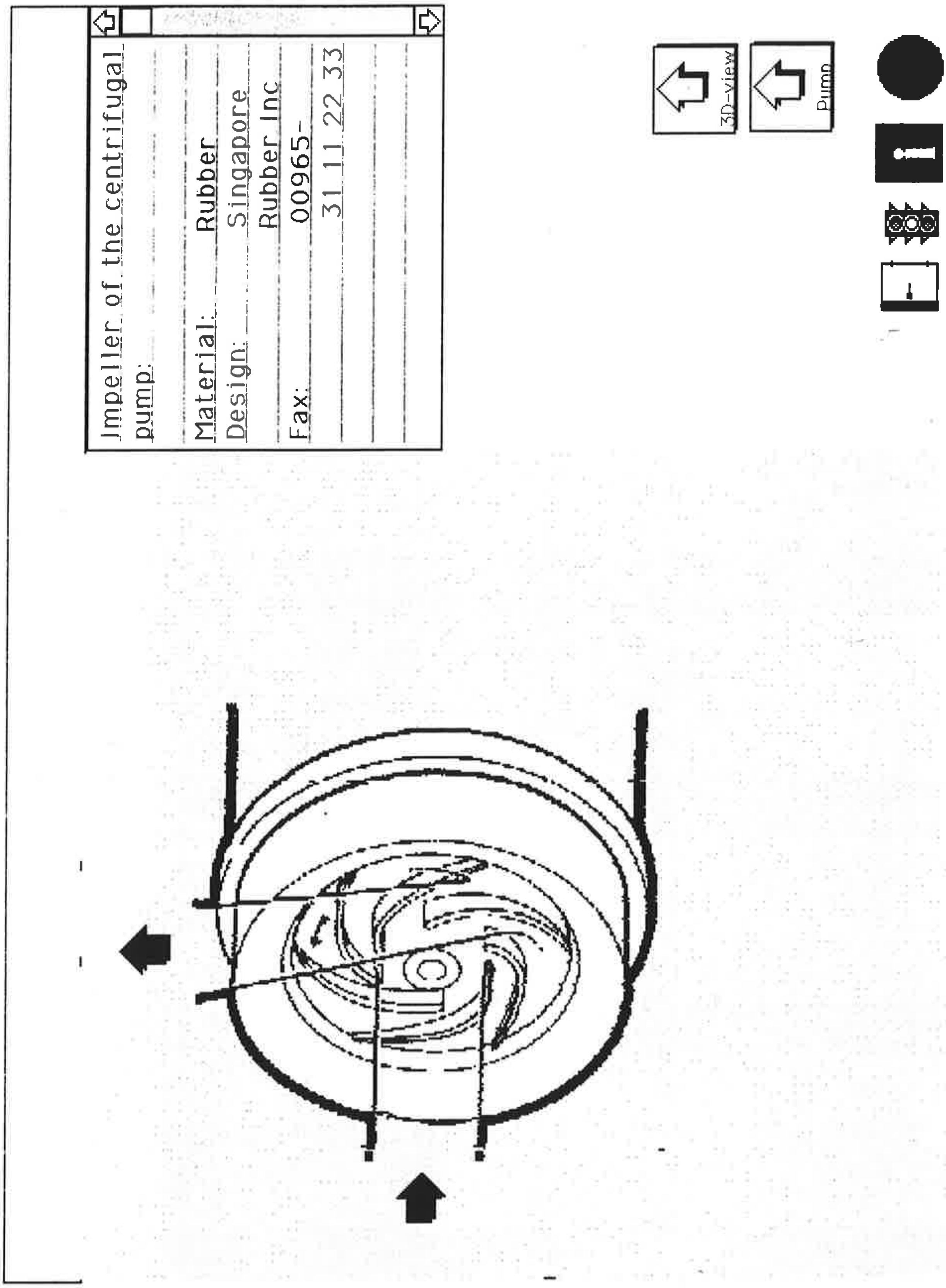
App B: The 3D View - Temperature



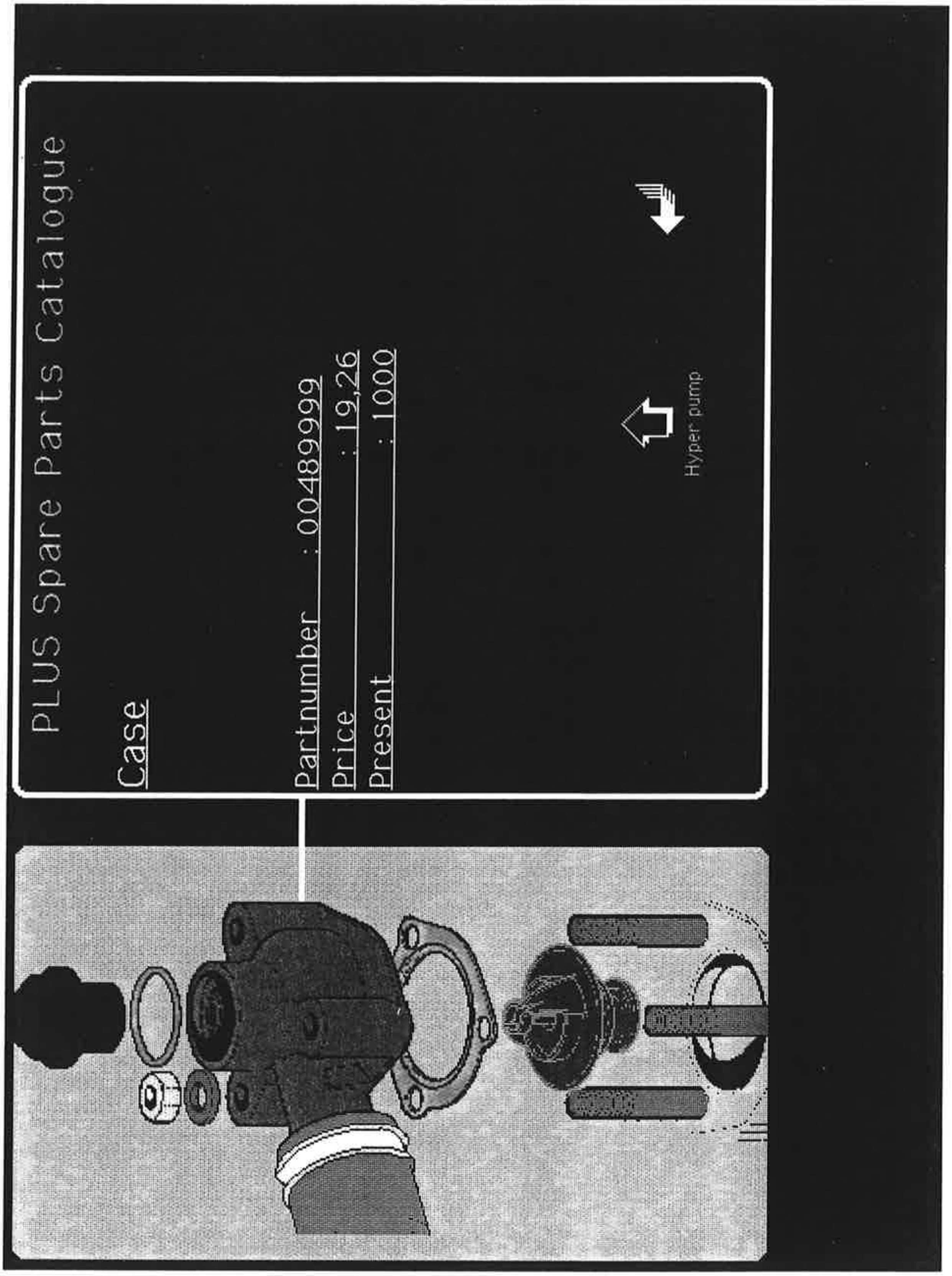
App B: The Pump View



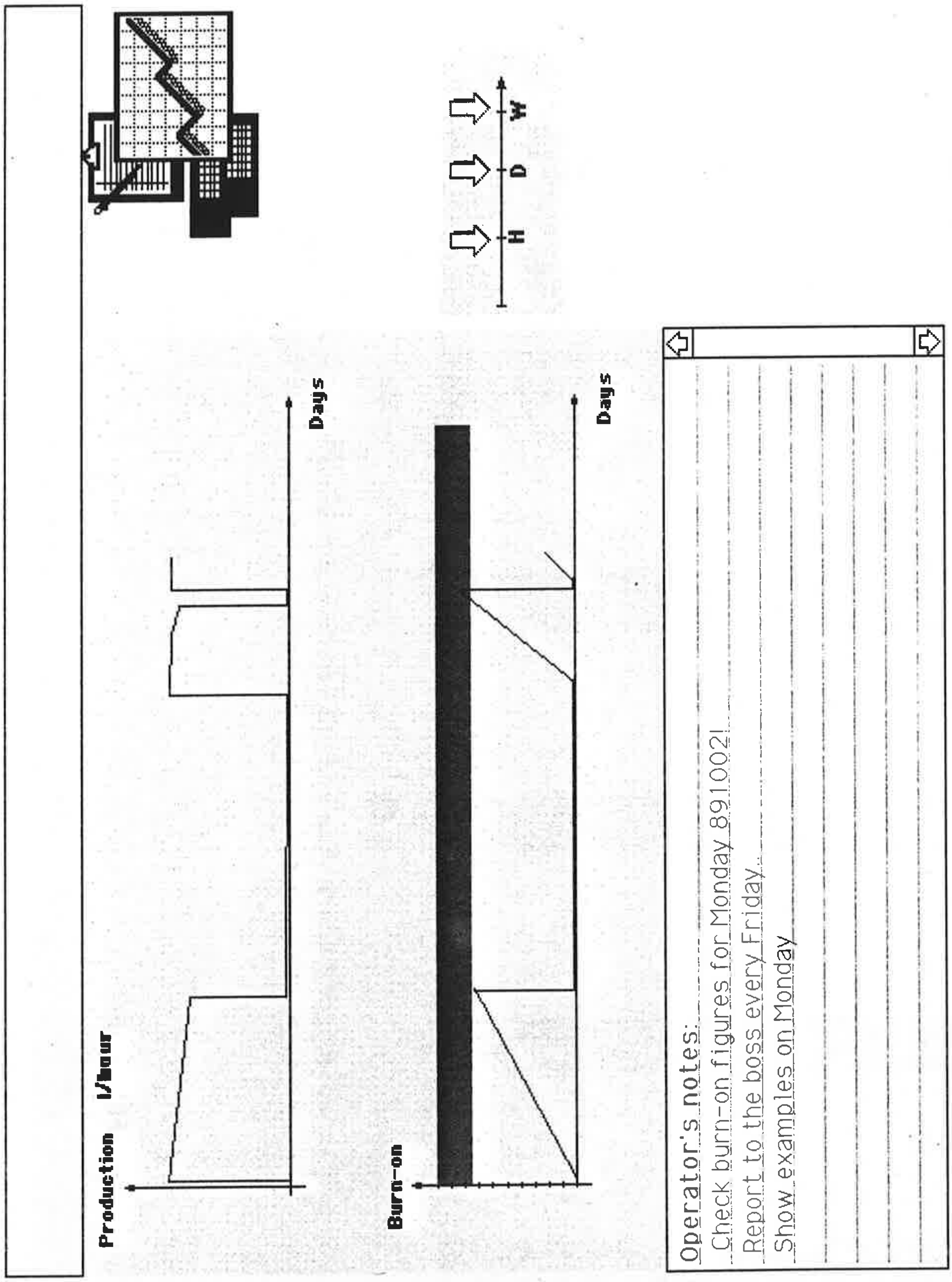
App B: The Impeller View



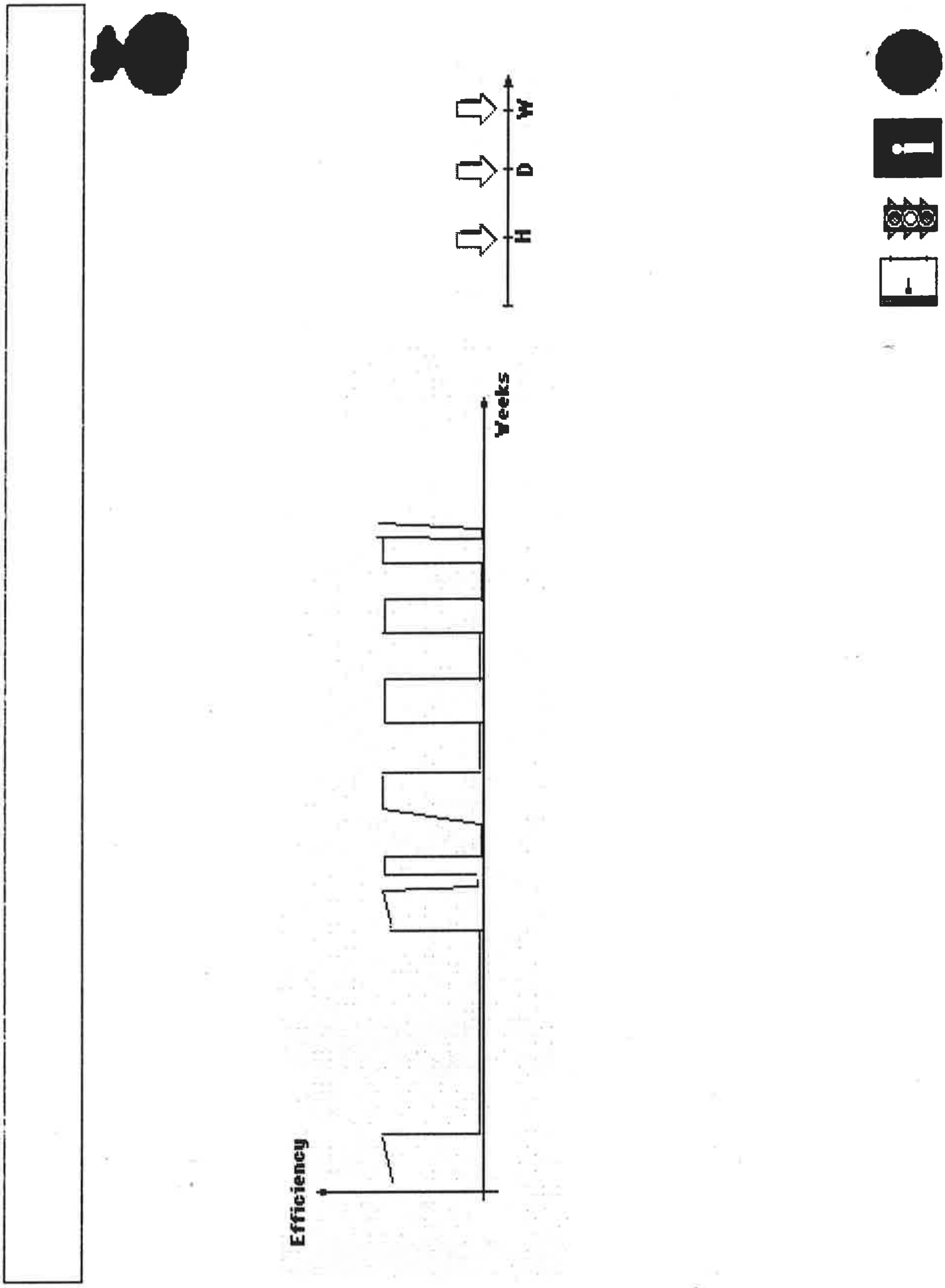
App B: The Spare Parts View



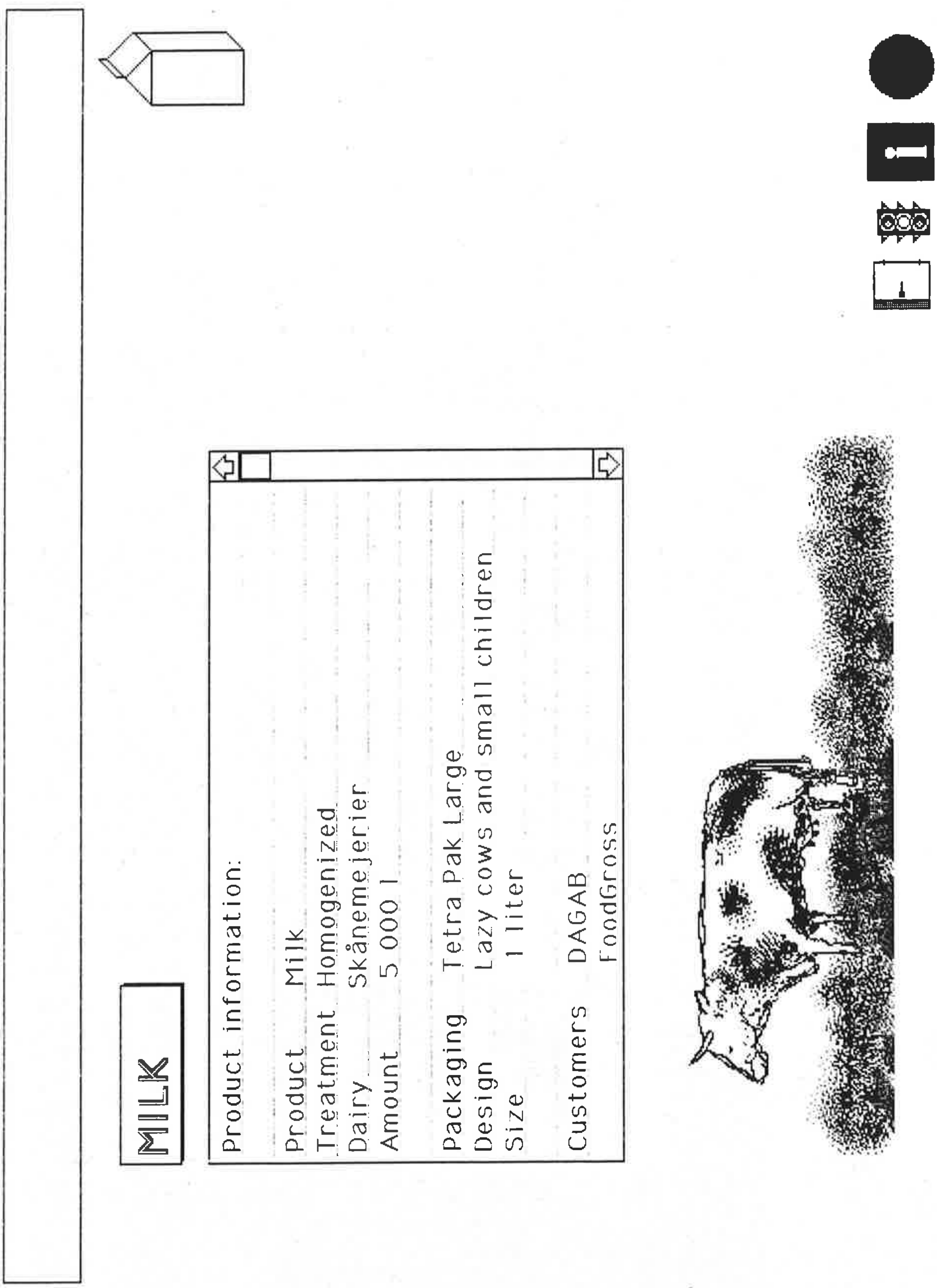
App B: The Trend Curve View



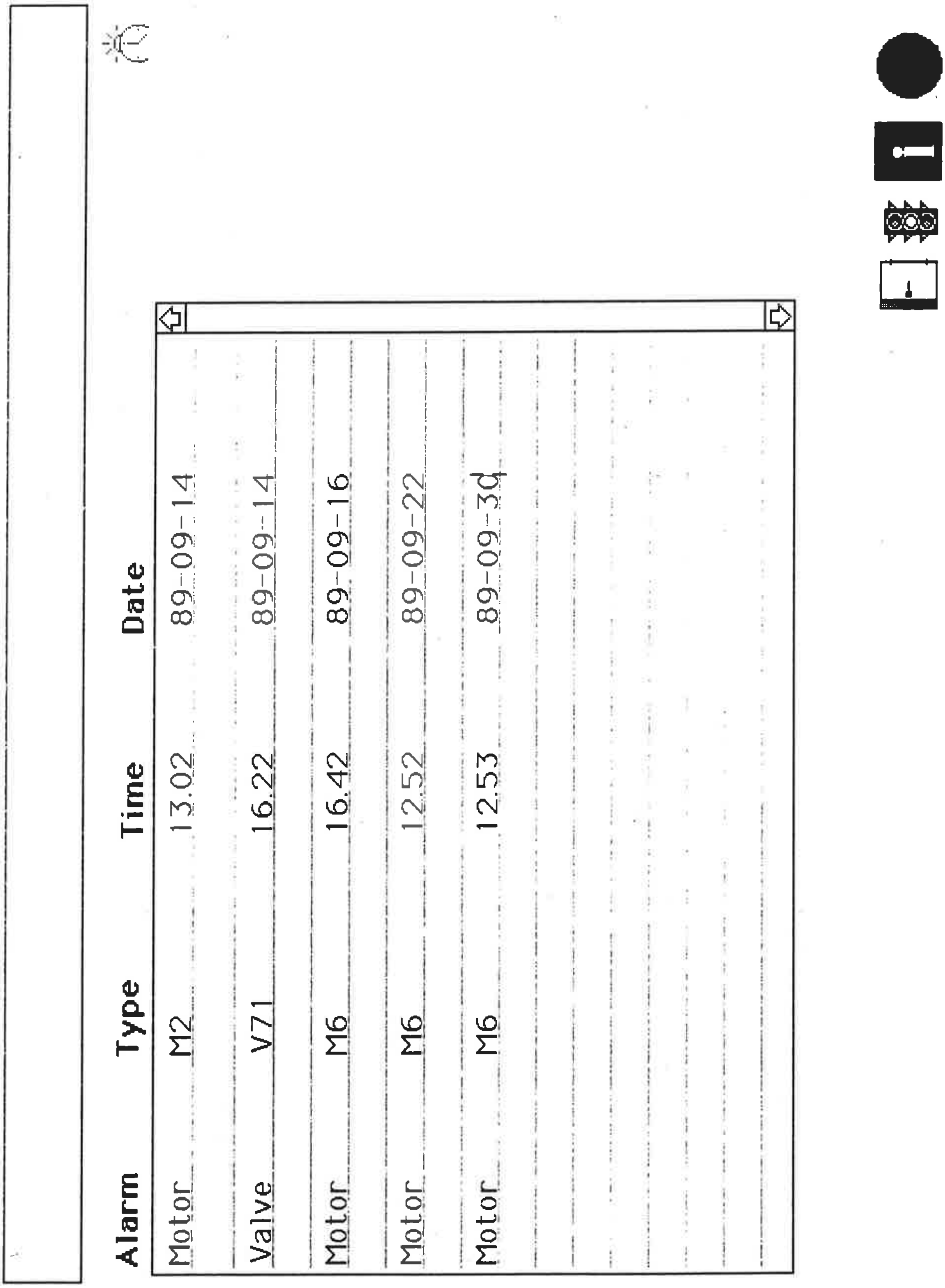
App B: The Efficiency View



App B: The Product View

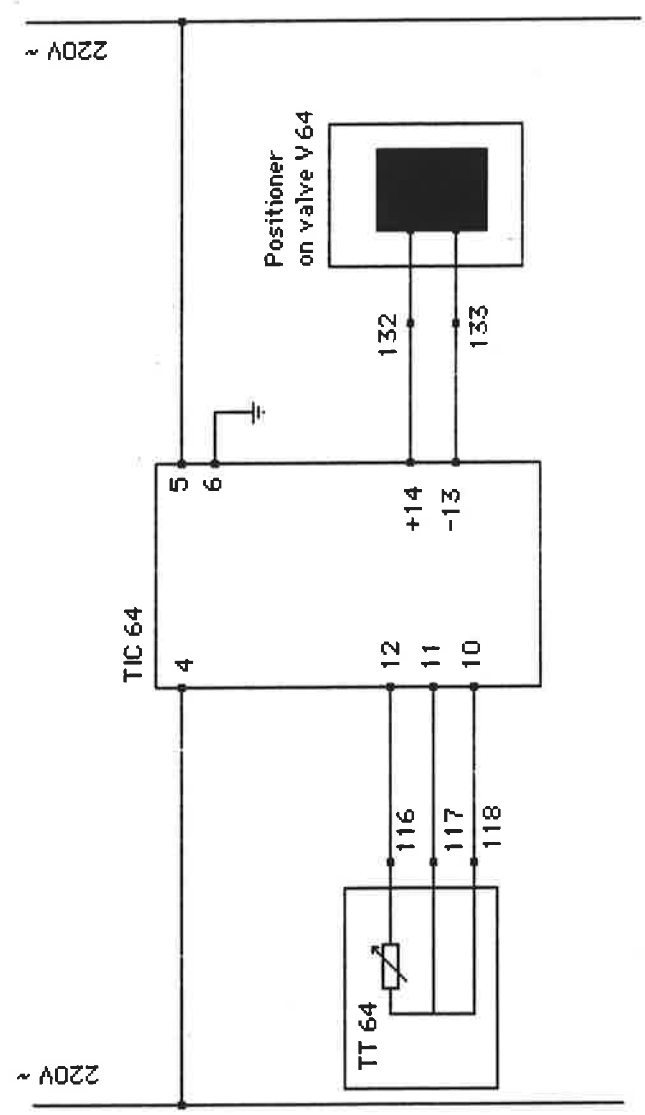


App B: The Alarm List View



Alarm	Type	Time	Date
Motor	M2	13.02	89-09-14
Valve	V71	16.22	89-09-14
Motor	M6	16.42	89-09-16
Motor	M6	12.52	89-09-22
Motor	M6	12.53	89-09-30

App B: The Electrical View



App B: The Positioner Circuit Photo View