ADMINISTRATIVE CONTROL IN THE SHIPYARD

Preprint of paper to be delivered at the ICCAS Conference. Tokyo, August 1973.

Professor Trygve Reenskaug
Central Institute for Industrial Research
Blindern, Oslo 3, Norway

{Scanned 2003-2-26 by Trygve Reenskaug. Some typos fixed, some probably added}

Key words: Production planning, production monitoring, shipbuilding, data processing, organisation, decentralisation.

ABSTRACT

The organisation of a yard may be strongly influenced by the information system it takes into use. The converse should also be true: Systems development should be subordinated the desired evolution of the organisation. The principle of Communicating Data Processes is introduced as a tool for building information systems that are easily moulded for fitting an existing organisation and for adaption to its continuous development. A description of the principle is given, as well as an outline of a new system designed for testing it out in practical shipbuilding.

1. INTRODUCTION

The modern ship is an extremely complex product requiring the co-ordination of a great variety of skills, materials and machines for its production. The good management of a shipyard therefore presupposes advanced tools for the planning and monitoring of production, and requires the consideration of economic, technological and social factors.

Information systems for various purposes are becoming important tools for shipyard management. It is our firm belief that there is strong interaction between an information system and the organisation it serves. As long as we want to maintain a conscious development of our organisation, we have to be very careful in introducing new information systems where organisational consequences are not well understood.

In this paper we shall present a general framework within which we intend to build various data processing systems for administrative purposes. The framework is based upon the principle of Communicating Information Processes. Its main purpose is to enable us to mould a new information system to fit an existing way of organising things, the resulting system being sufficiently flexible to be easily adapted to changes in this organisation.

Our major requirements to systems developed within the framework are:

1) We want to build man-machine systems. Manual override on automatically produced results must be a routine matter, and the systems must be able to compute the consequences of such overrides.

2) Responsibility, authority and competence will be spread out among groups

of people in the shipyard. Our total data processing system must be divided into subsystems that are attached to the various areas of responsibility. The subsystems shall be controlled by these groups and further developed by them.

3) A subsystem must be transparent to its responsibility group, so that its owners may fully understand their system's behaviour.

4) All subsystems must be open-ended, so that they may be attached to other subsystems belonging to the same or other responsibility groups.

5) The total system must be capable of continuous growth, it must be possible to add new subsystems or change old ones without disturbing the total system. It must be possible to include "foreign" systems developed by other people without upsetting the total system. The painful transition from one "generation" to the next can no longer be tolerated.

2. COMMUNICATING INFORMATION PROCESSES

Systems engineers have for a number of years been developing systems for handling separate functions. Such systems integrate a particular function throughout the company. One example is computer-aided design: Fig. 1 shows a simplified picture of Autokon, that spans the design function from the Project stage through Classification, Steel design and Lofting to the manufacturing of parts (1). This example illustrates that functionally oriented systems may become very large. Other systems, for example MAPLIS for materials planning, are still expanding and may become equally comprehensive (2). In time, the yard will be utilising a number of such systems, and the need for some sort of integration is growing. Today, functions are manually linked together in the various departments, each department being responsible for parts of several functions. Integration of each function is by formal and informal communication between departments.

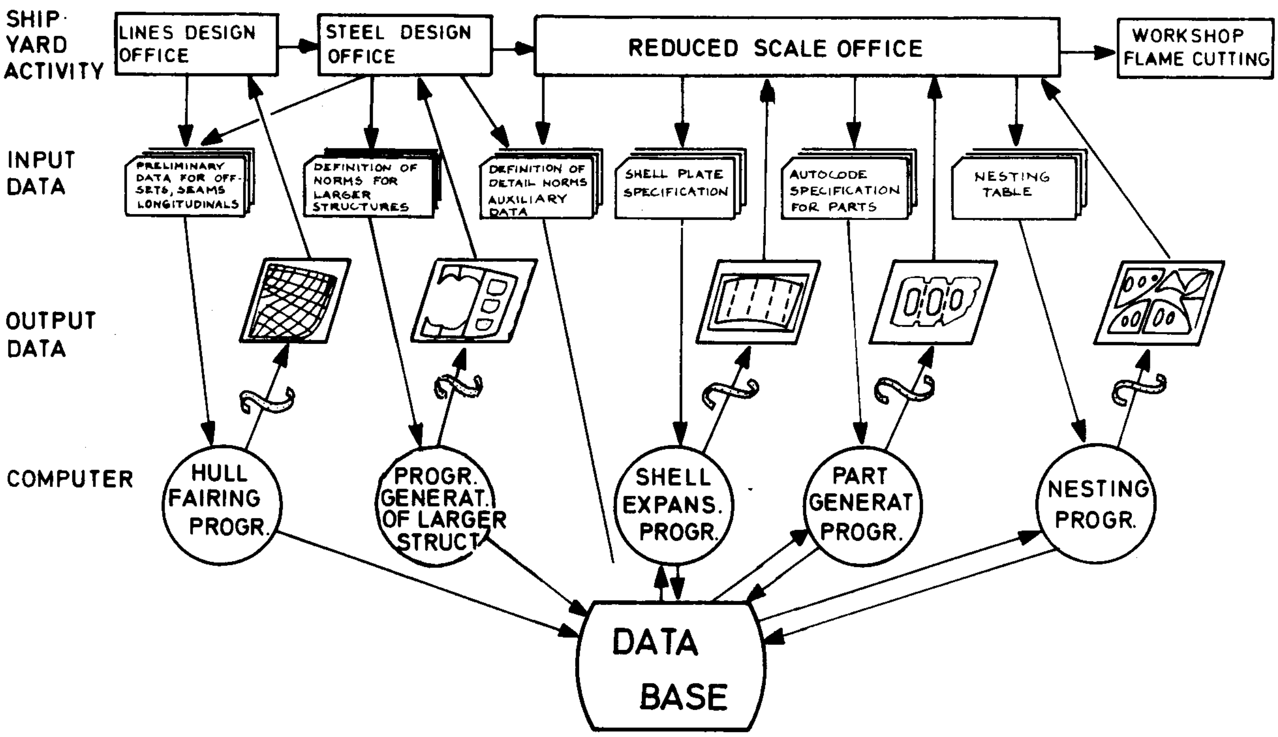

FIG. 1. INTEGRATION ALONG THE DESIGN FUNCTION (AUTOKON)

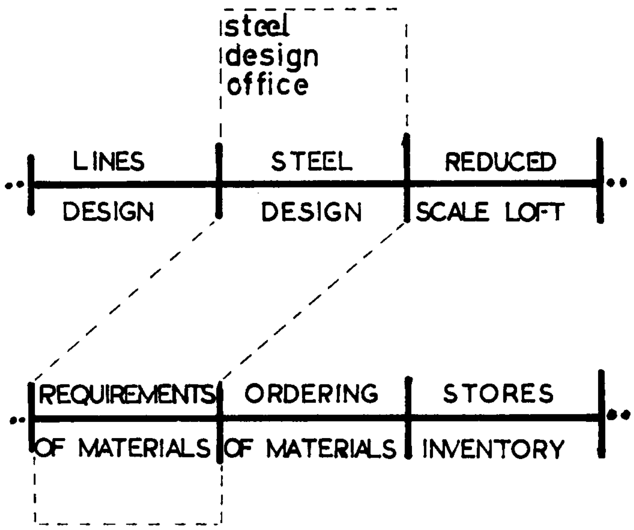

FIG. 2. A PART OF THE DESIGN FUNCTION IS LINKED TO THE MATERIALS' FUNCTION THROUGH THE STEEL DESIGN OFFICE PREPARING A LIST OF THE REQUIRED MATERIALS

We want our data processing system to support both the linking of functions within a department, and the integration of function parts between departments. Further, we want to be able to do this without losing the advantages gained by viewing each function as an integrated unit. We thus want to organise our total system in a matrix form. In this matrix, functions run as efficiently as possible along one axis, while the autonomy of the organisational units is maintained along the other axis. (See Fig. 2.)

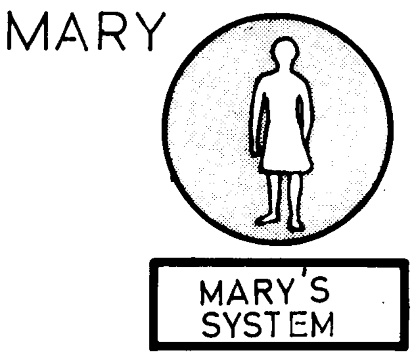

FIG. 3. A DATA SYSTEM AS SEEN FROM ITS USER

A large system which is built around these concepts may be viewed from various angles. We may study it from the functional viewpoint, and will find a system very similar to the functional systems of today. Alternatively, we may study the system from the individual users' point of view. We will then find a subsystem that is his personal system, its purpose being to aid him in his work. (See Fig. 3.)

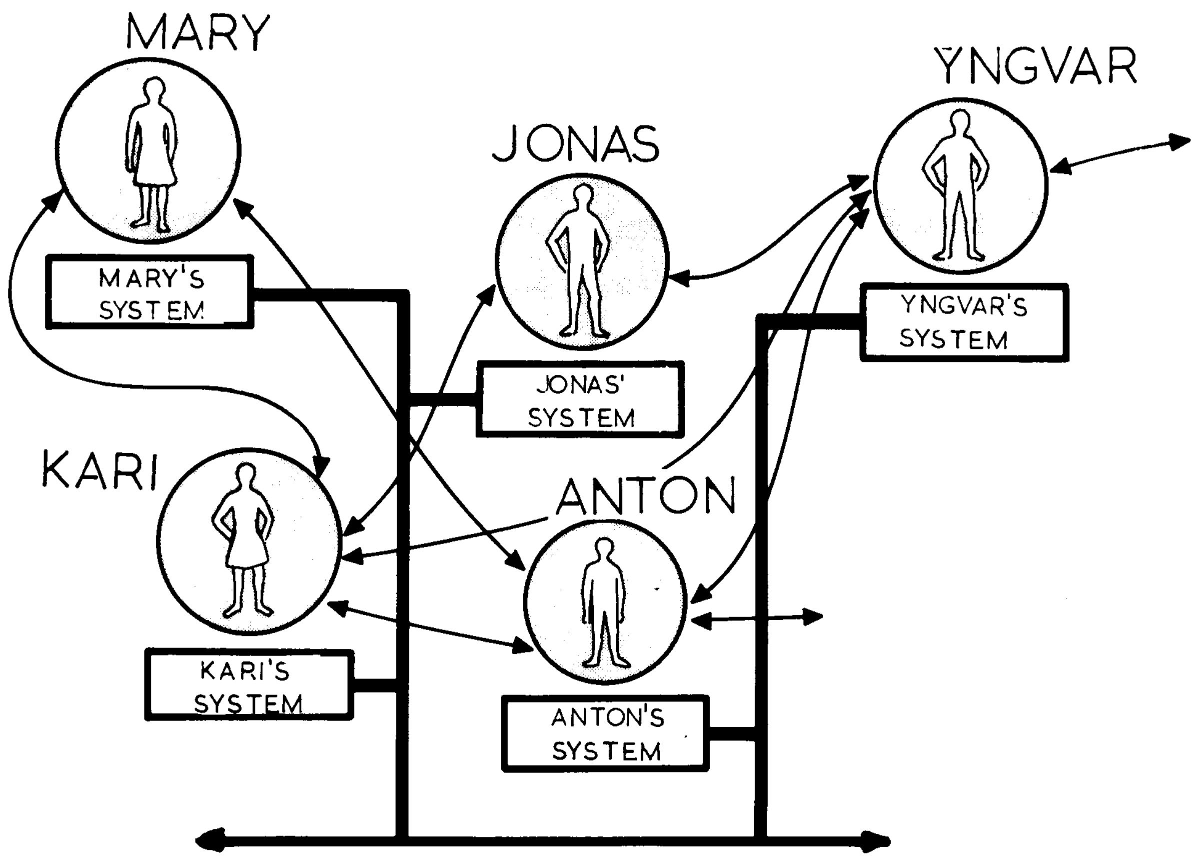

FIG. 4. COMMUNICATION WILL BE THROUGH THE TRADITIONAL CHANNELS AND THROUGH THE NEW COMPUTER SYSTEM

Our aim is that all users shall have such a subsystem at their disposal. By interconnecting these subsystems, we will get the additional benefit of an immensely powerful channel of communication. (See Fig. 4.) The communication through these channels must, however, be controlled by both sender and receiver. Otherwise the individual may lose the authority over the happenings in his own subsystem, and the close correspondence between the data processing system and the organisation it serves, will be lost.

We can now sketch the principle of Communicating Information Processes. A total system will be decomposed into a number of <u>components.</u>

Each component will be responsible for a well-defined part of the total job to be done. The component will include the data and instructions (know-how) needed for performing its job, and it may communicate with other components by sending and receiving messages. A component may be manual, automatic or a combination of the two. A process represents the part of a component that exists in a computer. This concept is defined in Section 4.1.

The components may be hierarchically organised, so that a component is controlled by its superior component. The top of each hierarchy is connected to the department in the yard which is responsible for that set of components.

Without regard to the hierarchical ordering, any component may send a message to any other component, but since this communication is controlled by both sender and receiver, it is in reality controlled by pertinent units of responsibility in the yard.

Fig. 5 illustrates how this permits us to integrate our data processing system for each function, without disturbing the normal lines of responsibility and authority in the yard.

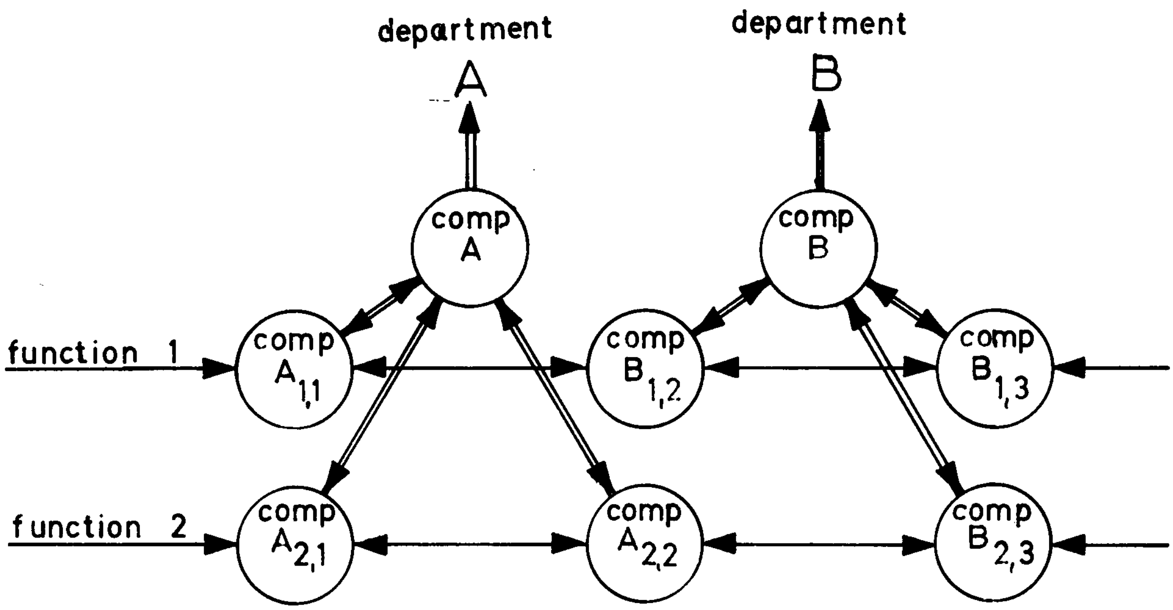

FIG. 5. TWO DECOMPOSED FUNCTIONS. EACH COMPONENT IS CONTROLLED BY THE RESPONSIBLE DEPARTMENT.

3. OUR RESEARCH PROGRAM

We have established a research program for realising systems according to our general principle. We expect this to be a long-term endeavour, and will therefore split the program into a number of projects. Each project will hopefully lead to a result which is useful in itself, and at the same time representing a step towards our general goals.

The project results may be of many kinds: General methods, insight into the operation of shipyards, special hardware or software needed for some class of applications, the realisation of a particular application in the yard, etc. For all projects, however, the quality of the results will be measured by their utility in practical shipbuilding, because this is our primary field of interest.

The probably most important class of problems we have to tackle, relates to the various functions of administrative control, such as strategic planning, control of materials and production monitoring. We will, of course, build on previous work done both by ourselves and by others, but also plan a sizeable effort within the present research program.

Secondly, there are a number of important problems associated with the future organisation of a yard: Questions of distribution and co-ordination of responsibility, authority and skills, and of how the various departments, sections and groups shall interact in order that the total yard may be running smoothly and each job be as satisfactory to the individual as possible.

Thirdly, there are questions associated with developing information systems. Our aim is to enable the yard of tomorrow to exploit the data processing techniques of tomorrow, and we will develop the hardware and the program modules needed to make this exploitation possible.

This research program may seem ambitious, and it is. But it should be remembered that we do not aim at perfect solutions or at solving all problems within our general framework. On the contrary, we will be rather opportunistic, trying to solve the pressing problems of modern shipbuilding in a way that will fit in with the expected future developments.

4. A SYSTEM FOR PRODUCTION CONTROL: PROKON

We have built a computer program for enabling us to experiment with the principles of communicating information processes, and are now using it for experimental planning in close co-operation with the planning department of Stord Verft. Our first version of the computer program is called Prokon-0. This program enables the computer to interpret instructions written in the Prokon-0 language. This is an ad hoc language selected for minimum cost of implementation. Prokon-0 is based upon Alkon (1), which is the plane geometry program of Autokon, and most features of Prokon-0 will also be found in Alkon.

The zero in Prokon-0 indicates that its main purpose is experimentation. It is our intention to build a Prokon-1 based upon the experiences with Prokon-0. The Prokon-1 language will be more problem-oriented, and will contain a number of features not existing in the experimental version.

Prokon-0 is a batch program in the sense that each run on the computer is initialised, the input cards are read in and acted upon, and the results are printed out. At the end of the run, however, the state of the system is stored in Prokon's own Data Base, so that the next batch run will be a direct continuation of the previous one.

The central concepts of Prokon-0 are the concepts of Processes and Communication.

4.1 Process

Let us imagine a bounded set of storage locations in a computer. The locations contain data and instructions (procedures). The procedures may operate upon the data so that the contents of the storage locations vary with time. This variation with time we call a process.

A process is born, lives its life and dies. It is thus dynamic. We may at any point in time take a snapshot of the process. This snapshot will show the state of the process at that time. A process snapshot is thus static.

4.2 The Birth of a Process

Processes may give birth to new processes. The mother process builds up the required locations within herself, and assigns initial values to these locations. At birth, a special command detaches these locations from the mother process and gives them their own name. The life of the new process has then begun.

4.3 Communication

The processes communicate by sending messages to each other. This is the only means of communication between processes. A message may consist of any number of integer values. In theory, it is sufficient that sender and receiver agree upon the significance of the data in the message. In practice, we have found it useful to standardise messages, so that a message of a certain type has a fixed significance for any sender and receiver. The following information is associated with a message:

Sender's name
Receiver's name Message type
The integer number of the message itself

Processes are passive most of the time, residing in the Data Base. A process is activated when it receives a message. It is then moved into core, and one of the procedures of the process is executed.

The choice of procedure is determined by table look-up. Each process contains a double-entry table showing the correspondence between the type of an incoming message and the name of the procedure that will handle it. Since each process has its own table, a given message may be handled differently in different processes. This corresponds to the manual transmission of messages within an organisation: Both sender and receiver must be able to interpret the message, but it is the receiver who decides what to do with it.

Prokon-0 is running on a computer with only one processor. Only one process may therefore be active at a given point in time. This active process will be left running until termination of the message-receiving procedure. The procedure may of course give birth to new processes, and it may send out any number of messages. All messages will be queued up by the Prokon-0 system. When the active process has terminated, the system will select a message from its queue and activate the correct procedure in the receiving process.

4.4 Component

In the chapter on Communicating Information Processes, we used the term “component". In our data system, components will be represented by processes. We expect our data system to contain additional processes that are not recognised as separate components by the outside world. The word "component" is thus oriented towards the organisation in the yard, while the word "process" is oriented towards the computer.

5. PILOT PROJECT: EXPANDED NETWORK PLANNING

Stord Verft, the largest yard in the Aker Group, is the pilot yard for our work with administrative control systems. This yard has for a number of years been planning their hull assembly and outfitting activities with network techniques. They have for some time been feeling the limitations of these techniques for handling the great variety or conditions and restraints round in shipbuilding.

Our first application project is to build a new system for the planning and monitoring or hull assembly and outfitting. The system will initially be based upon a relatively simple model, that will be continuously improved as the yard gains experience with the new tool.

In the model, the yard with its operation is decomposed into a number of components. Each component represents some important part of the whole: Some components representing materials, some representing resources and some representing the product and the operations that have to be performed in order to produce this product.

In accordance with the requirements set out in Chapter 1, we want the different components to be controlled by the departments responsible for that particular part of the yard. Fundamentally, the actual planning will take place within the departments and by communication between them, but the tedious and repetitive part of the planning will occur automatically within processes in the computer and through communication between them.

Our total planning model will be a complex one, but we think of each component in a simple way: We imagine a component to behave like a person who has well-defined goals and field of responsibility and who ensures that his goals are met through communication with others. For example, a component being responsible for the manufacture of a certain building block by a certain date must reserve the necessary resources for the work and must make sure that the required materials are available. Similarly, a component responsible for a particular resource must plan the utilisation of it in a manner consistent with the considerations which apply to that resource.

We do not intend to build such a total model at once. On the contrary, we will start as simply as possible and then let the model evolve as the yard finds it desirable. The model we intend to start with, is a model very similar to the network model the yard employs today. Our first job is therefore to model a network by means of Communicating Information Processes.

5.1 An Alternative Network Model

We will let each network activity be represented by one component. The actual planning will take place in three stages: Frontloading, backloading and computation of slack.

Each activity component must at least contain the following information:

The names of its predecessor(s)
The names of its successor(s)

- Procedure for frontloading
- Procedure for backloading
- Procedure for computation of slack
- Time and resource requirements
- Earliest start and finish (after frontloading)
- Latest start and finish (after backloading)
- Free and total slack (after computation of slack)

In frontloading, each activity is done as early as possible. The following message will be used for this part of the planning:

Type: Frontloading
Message: Predecessor's name and earliest finish time

Initially, such a message is sent to all start activities, the time being the earliest starting time for the complete network.

An activity component will expect one such message from each predecessor. The procedure handling the front loading message will therefore just remember the incoming message until messages have been received from all predecessors. (See Fig. 6.) The activity component will then select the latest incoming finish time as its earliest starting time. On the basis of the activity's time and resource requirements, its own earliest finish time may be produced. Frontloading messages may then be sent to all successors. The frontloading thus works its way through the network, leaving behind information about early start and finish times in every activity component.

It is a procedure residing in the component itself that produces the earliest finish time when the earliest starting time is known. For some activities, this production may be a simple sum based upon a fixed activity time. For others, it may require communication with production resource components or material's components. In some cases, it might even be convenient to use subnetworks. The method thus gives a considerable flexibility in the method of planning.

Backloading and computation of slack follow the same pattern, and will not be discussed here.

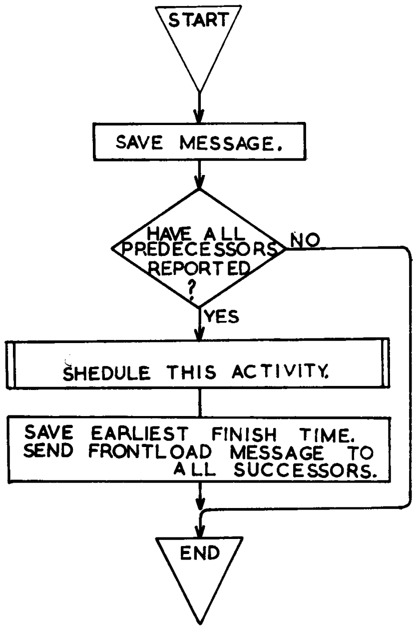

FIG. 6. TYPICAL FRONTLOADING PROCEDURE. These procedures may be different for the different activities in the same network. For some activities, the scheduling may be based upon a fixed activity time. For others, it may involve communication with the relevant resource components, and in some cases it may consist of the front- loading of a subnetwork.

6. CONCLUSION

Our starting point was the requirements of a real shipyard as we could see them. These requirements led to the formulation of a basic tool in data processing: The principle of Communicating Information Processes. This principle will first be applied to today's most pressing set of problems in the yard: The detailed planning of hull assembly and of the daily monitoring of production. The result of this development will be a system succeeding several older systems in the yard, for example their network planning system.

We expect the new planning system to be a valuable result in itself. It is, however, only a specific example of a number of systems that may be built using the principle of Communicating Information Processes. We are presently investigating the use of Prokon-0 in such diverse applications as the computer-aided design of electronic circuits and in studying the effects of pollution on our environment.

Finally, we would like to stress that we view the principle of Communicating Information Processes as a tool to be used together with the other tools of data processing. The key factor for further development in the shipyard will still be the individual persons working in the yard, their competence and their engagement in the advancement of the yard.

7. ACKNOWLEDGEMENTS

The Central Institute for Industrial Research and the Aker Group have for a number of years been co-operating in the development of advanced digital systems. This co-operative effort is being governed by a joint body called SIAG. The work reported in this paper is part of the SIAG development program.

REFERENCES

1) Anton Landmark: ALKON - A LANGUAGE FOR ALGORITHMIC DESIGN. ICCAS Conference, Tokyo 1973.

2) Ragnar Sæveraas: MAPLIS - A Tool for Material Planning and Control in the Shipbuilding Industry. ICCAS Conference, Tokyo 1973.