The development and implementation of an information flow monitor system

Computergesteund ontwerpen en fabriceren van discrete produkten

The development and implementation of an information flow monitor system

Ir. B.G.J. Wijers WPA 1367. aug '92

Wijers, B.G.J.

The development and implementation of an information flow monitor system / B.G.J. Wijers. - Eindhoven : Instituut Vervolgopleidingen, Technische Universiteit Eindhoven. - I11.
Final project of the study Computer aided design and manufacturing of discrete products. - With ref.
ISBN 90-5282-213-1
Subject headings: Information flow monitor systems.

Summary

This report describes the development of a monitor system for the company TRAUB AG. The development of this system has been executed as the final project of the study Computer Aided Design And Manufacturing Of Discrete Products.

The company TRAUB AG is currently working on a project with the aim of creating a computer system, which integrates the information processing of the design, the planning and the production departments. This project is partly in cooperation with the institutes FHG/IPK (Berlin) and TNO-IPL (Eindhoven).

At this moment different departments within the company use several computer systems for partial solutions in the information processing. There is little or no integration between these systems. This results in idle-time of production machines and late deliveries to clients. Within the company the insight fails in the exact causes of these problems. In order to get an overview of the causes of the problems, a monitor system of job and resource information has been developed. The aim of this system is to produce up-to-date overviews of the states of all resources in the production process and of the progress of the jobs within the production department.

In order to meet the objectives, the first step has been the analysis of the information flow between the planning department and the production department. On the basis of this analysis a framework for the monitor system has been designed. Portions of this framework have been implemented with the help of a commercial software package. As an extension of this package extra processes have been developed for the communication between the monitor system and the production machines, and for the creation of an overview of the job progress and the state of the production machines.

The resulting system has been operating for several weeks. The success of the system is limited due to major problems with the used commercial software. Not withstanding this setback, valuable insights have been generated, concerning the continuing development of the monitor system. Most of the problem areas have been found and will be solved in a later stage.

Summary 1
1. Problem description 3
1.1. Introduction 3
1.2. Present situation 4
1.3. Objectives 6
2. Analysis 9
2.1. Factory structure 9
2.2. Job structure 10
2.3. Object definition 11
2.3.1. Manufacturing part 11
2.3.2. Master Job 11
2.3.3. Detail job 12
2.3.4. Program 12
2.3.5. Production machine 12
2.3.6. Pallet 13
2.3.7. Cutting tool 13
2.3.8. Auxiliary tools 13
2.4. Necessary object information 14
2.4.1. Planning department 14
2.4.2. Cutting tool department 14
2.4.3. Auxiliary tool department 15
2.4.4. Programming department 15
2.4.5. Production department 15
2.5. Available object information 15
3. Framework 18
3.1. Hardware configuration 18
3.2. Data objects 19
4. Implementation 22
4.1. Host 22
4.2. Production computer 22
4.2.1. FREENET processes 22
4.2.2. Monitor process 24
4.3. Production machine unit 25
4.3.1. FREENET processes 25
4.3.2. Database 25
4.3.3. Terminal handler 26
4.3.4. LSV2 processes 28
4.3.5. V24 handler 32
4.3.6. Digital data acquisition 32
5. Results 34
5.1. Terminal handler 34
5.2. Communication processes 35
5.2.1. Digital data acquisition 35
5.2.2. V24 handler 36
5.2.3. LSV2 processes 36
5.3. Monitor process 37
5.4. Tool handling 38
6. Conclusions 40
Reference list 41

1. Problem description

1.1. Introduction

The subject of this report is the development of a monitor system for the company TRAUB AG. The development of this system has been executed as the final project of the study Computer Aided Design And Manufacturing Of Discrete Products.

Founded in 1938 to produce curve controlled lathe machines, the company TRAUB AG currently produces curve controlled lathe machines, CNC-lathe machines, CNC-manufacturing centers, and teaching software. In 1990 this company had a turnover of 450 million D-mark and 2600 employees in Germany, France, Italy, and Brazil.

The company is currently working on a project with the aim of creating a computer system, which integrates the information processing of the design, the planning and the production departments. This project is partly in cooperation with the institutes FHG/IPK (Berlin) and TNO-IPL (Eindhoven).

At this moment different departments within the company use several computer systems for partial solutions in the information processing. There is little or no integration between these systems. This results in idle-time of production machines and late deliveries to clients. Within the company the insight fails in the exact causes of these problems. In order to get an overview of the causes of the problems, the first step in the project is the development of a monitor system of job and resource information. The aim of this system is to produce up-to-date overviews of the states of all resources in the production process and of the progress of the jobs within the production department. The resulting data will be used to detect problem areas with the help of statistical analysis over certain time periods. In a later stage this monitor system will be incorporated in a production planning system, which will be developed on the basis of the acquired data.

The main objective of the production planning system is to find the best way to manufacture a given set of products within the constraints of the manufacturing resources. These constrains are the limited availability of production machines, tools, etc., and the delivery date of the finished product. The planning system generates lists of jobs, defining the jobs for the machine groups in the production department. Those lists are called the master job lists of the machine groups. Because of the interdependency of the jobs, it is rather difficult to determine the best distribution of the jobs over the available resources in the production department. Moreover, the distribution problem is increased by the existence of rush orders and the occurrence of breakdowns of resources needed for the execution of a job. Usually the planning system can not solve the problems by simply rearranging the master job lists. The necessary reallocation of jobs to other production machines can require several planning sessions. The higher the capacity utilization of the resources, the higher the risk a small disturbance will propagate and cause many additional problems. A quick analysis of the situation and creation of alternatives will limit the spread of the disturbances. When generating alternative master job lists, the planning system must have access to the data reflecting the availability and the actual workload of the resources.

The link between the planning department and the shop floor of the production department is essential. The feedback of information from the shop floor, with respect to the state of the resources and the progress of the jobs, is vital for the creation of realistic master job lists. On the other hand, it is important for the shop floor to receive all necessary data needed for the execution of the master job lists. In this case, the monitor system has to be able to transfer data in both directions.

1.2. Present situation

In the present situation the basis for the existing production planning system is the year-production plan. The year-production plan describes the quantity and the types of machines, which will be produced in the following year. Once a year this plan is generated based on the production in the previous year, the expected orders, and the affirmed orders for the following year. During the year, this plan changes depending on the arrival of client orders for specific machines. The client orders specify the types and extra options of series of production machines. If an order for a specific machine is received, the specifications of the ordered machine are compared with the machines in the year-production plan. If a machine, which has not been reserved for an other order, is comparable to the ordered machine, the machine is reserved for this order and the specifications are changed to meet the order. If no comparable machine is found in the year-production plan, the machine is added to the adapted plan.

This plan is the input for a production planning system. This so-called PPS-system has access to lists defining the decomposition of the machines in manufacturing parts and lists defining the manufacturing sequences and manufacturing times needed to produce a manufacturing part. As a first step in a planning session the PPS-system splits the machines in the current year-production plan into the manufacturing parts. In the next step, the PPS-system schedules these manufacturing parts backwards according to the necessary run-times and manufacturing-times in order to meet the required delivery dates. The PPS-system periodically generates a schedule for a ten-day period. This schedule consists of lists of jobs for the production department, detailing the manufacturing dates of the jobs and the machine groups executing the jobs. These lists are called the master job lists. While generating the master job lists, the PPS-system only checks the availability of the workpiece material. As soon as the workpiece material is available, the PPS-system releases a job for manufacturing. As the final client order is received only weeks for the delivery, the specifications of the planned machines are liable to changes. The client specified options often result in changes in the design of a machine and consequently to changes in the manufacturing sequences. This makes it necessary to reschedule the master job list.

A job in the master job list is a combination of manufacturing or assembling sequences on a number of identical manufacturing parts executed within one machine group. Within the company a machine group consists out of a group nearly identical production machines. The differences between the production machines are the reachable tolerances, as a consequence of the history of the machine, and the composition of the tool storage magazines. The manufacturing sequences in a job can require several programs. Each program corresponds to the manufacturing sequences of a manufacturing part positioned with the help of one fixture. When pallets are used for the transport of manufacturing parts within a machine group, a pallet contains only one manufacturing part. Within a machine group a job can be executed on several production machines. The Meister of the machine group has the authority to distribute the execution of a job over the machines in the machine group. The distribution is mostly based on the criteria to change the cutting tools in the tool storage magazine of the production machines as little as possible. Some possibilities of the allowable distributions are shown in figure 1. This figure shows several ways to manufacture a job of four manufacturing parts

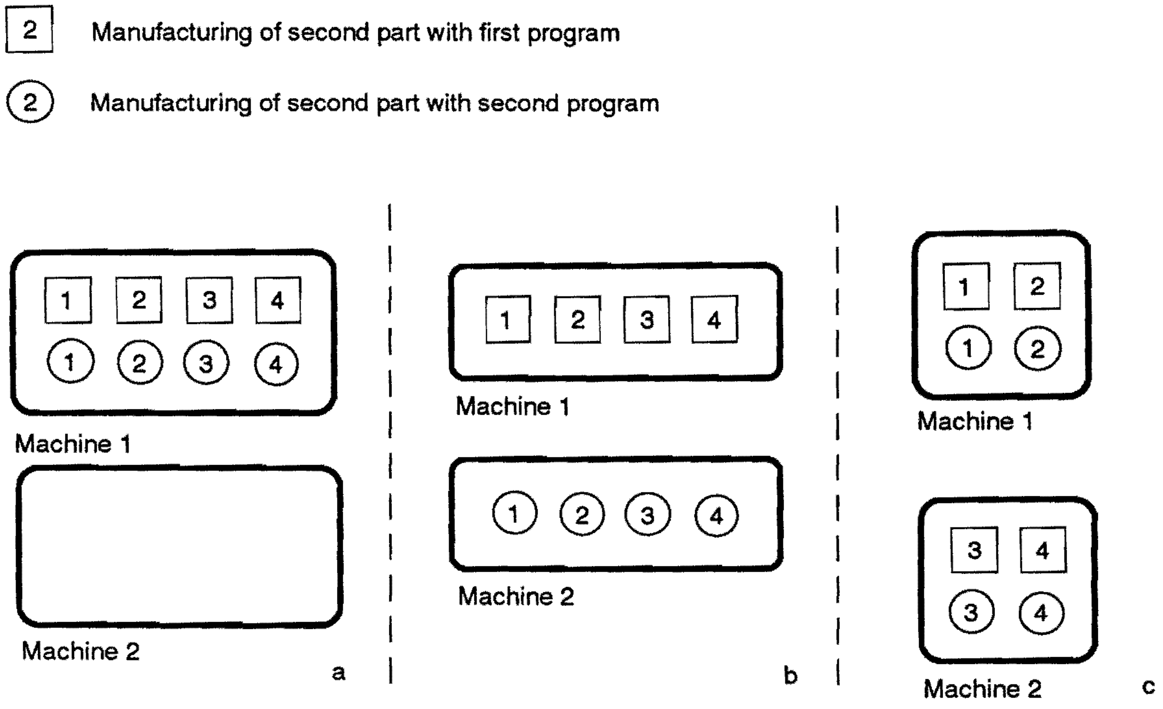

2 Manufacturing of second part with first program
2 Manufacturing of second part with second program
1 2 3 4
1 2 3 4
Machine 1
Machine 2
a
1 2 3 4
Machine 1
1 2 3 4
Machine 2
b
1 2
1 2
Machine 1
3 4
3 4
Machine 2
c

Figure 1: Several ways to manufacture a job of four manufacturing parts requiring two programs within a machine group containing two production machines

requiring two programs within a machine group of two production machines. The Meister can alter the execution dates of the jobs in the master job list as long as the end-delivery dates are not surpassed.

The scheduling of the jobs is liable to changes caused by changes in the master job list, unplanned rush jobs, the absence of resources or personnel, the existence of scrap and rework parts, and the failure of production machines and resources. This makes it difficult to create a structured job stream. Currently the progress of the jobs is followed by means of terminals. Job start, end, and interruption have to be reported at these terminals. For this purpose each batch of manufacturing parts is accompanied by a document describing the required jobs. The jobs in the document are labeled with a bar code. When the machine operator has to report changes to the state of a job, the operator reports the change of a state with the help of function keys on the terminal, and identifies the job by reading the bar code in the document with a light pen. A problem with this system is, that data on the progress of a job is only available at the specified moments. As a result, there are no possibilities to detect for example the cause for a delay in the planned start time. Therefore, the planning department has no possibilities to do preventive rescheduling. Another problem is caused by the absence of data on the real capacity use of the resources. On the basis of the data gathered by the terminals, an indication of the capacity use of a machine group can be deducted. The value of this data is suspect, because it is found, that operators report the end of a job after the elapse of the specified run-time of the job, even if the job is finished earlier or later.

All the production machines have a tool storage magazine. This magazine exists out of a section with non-alternating and a section with alternating cutting tools. The frequency a cutting tool is utilized over a certain time period, determines the composition of these sections. The cutting tools needed for a job, are specified inside the programs used in the job. As soon as a job is planned for a specific production machine, the required cutting tools are compared with the available cutting tools in the tool storage magazine. Differences are reported with the help of so-called cutting tool difference lists. The alternating cutting tools are exchanged depending on the availability of free places in the magazine and the expected use of the tools in following jobs. Cutting tools removed from the tool storage magazine, are moved to the cutting tool presetting room. If a cutting tool is used in one of the following planned jobs, it is stored in one piece. Otherwise the cutting tool is disassembled and the pieces are stored. The number of cutting tools in the production department makes it difficult to follow all tools. In a later stage the monitor system will be extended for this purpose.

1.3. Objectives

As stated before, the production flow depends on many different factors. The major factors are the availability of personnel, resources and materials. Because of the scarcity of these production factors the progresses of different jobs will be interdependent. Ever shorter run-times and smaller stocks of resources and materials enforce a quicker detection of problems in order to circumvent problems in the production flow by means of replanning. To make this possible the current state of every factor in the production flow has to be available. As a first step to reach this goal, a monitor system has been developed in order to follow the progress of the jobs and the state of the production machines. In a later stage the system will be extended to follow the state of the cutting tools and auxiliary tools. At this stage no requirements have been defined for those resources. This is due to the development of new handling methods for these resources. All the data generated by the monitor system has to stay available for statistical analysis. At this moment the requirements for this analysis have not been defined. In order to make a statistical analysis possible, all generated data is stored.

The most important detail in the progress of a job is the state of the job. The following states have been defined for internal use in the company:

- unknown: No valid data is available on the state of the job.
- waiting: The job is waiting for all resources to be available. All missing resources have to be listed.
- available: All resources are available.
- executing: The manufacturing has been started. The production machines, which are executing the job, have to be listed.
- interrupted: The manufacturing process has been interrupted. The cause of the interruption has to be listed.
- finished: The job is finished. At this moment it is unclear how to identify and report rework and scrap parts. Therefore this state signals, that all necessary manufacturing has been executed.

Other details of the jobs, that have to be reported with the help of the monitor system, are:

- manufacturing-time: The actual total time the manufacturing parts have been manufactured.
- run-time: The actual total time since the start of the job.
- starting-time: Planned start of manufacturing
- end-time: Planned end of manufacturing
- planned production: Planned number of manufacturing parts
- actual production: Actual number of finished manufacturing parts

The data has to be available for all jobs in the production departments and has to stay available for a limited time after the completion of a job.

One of the most important factors in the progress of the jobs is the state of the production machines. For an effective production plan it is necessary to have an overview of the actual utilization of a production machine and an overview of the average times needed for test-runs, maintenance and setting-up. In order to make this possible the following details have to be available for the production machine:

- job list: The state of all jobs planned for the production machine including the cutting tool difference lists.
- operation mode: The mode of operation of the production machine. Currently the following modes are considered: automatic, test-run, maintenance, and unknown.
- operating status: The operating status of the production machine. Defined are running program xx, failure, program xx interrupted, and setting up.
- alarm reports: Reports containing information about existing alarms. They contain the alarm number, the alarm message, the source (LSV2-communication, NC-control or PLC-control) and the art of the alarm (fatal or warning).
- tool storage magazine: The list of cutting tools currently in the tool storage magazine.

These details have to be available for the production machines, combined for different machine groups, and combined for the cost places. All the data has to be represented in a machine independent format. The differences in the availability and the format of the data received from different types of production machines, constitute a major problem. The data has to be transformed to machine independent formats and the unavailable data has to be generated manually by the machine operator with the help of terminals.

The system has to be as automatic as possible, for example automatic downloading and uploading of programs at the start and end of a job. The main reason for this is to circumvent possible errors and inconsistencies as much as possible. In the present situation it happens, that an operator reports the end of a job after the elapse of the planned run-time, even if the job is not yet finished, or has been finished beforehand. On the other hand, at a breakdown of a production machine the operator hopefully will limit the damage first, before signaling the interruption of a job. Both actions are sources of inconsistencies within the monitor system. At those points, where the machine operator is needed, the necessary actions of the machine operator have to be as easy and as simple as possible (selection instead of typing, good layout) in order to circumvent mistakes and avert resistance to the system.

In order to meet the objectives, the first step has been the analysis of the information flow between the planning department and the production department. On the basis of this analysis a framework for the monitor system has been designed. Portions of this framework have been implemented with the help of a commercial software package. As an extension of this package extra processes have been developed for the communication between the monitor system and the production machines, and for the creation of an overview of the job progress and the state of the production machines. The initial results will be discussed with recommendations and conclusions about the implemented system.

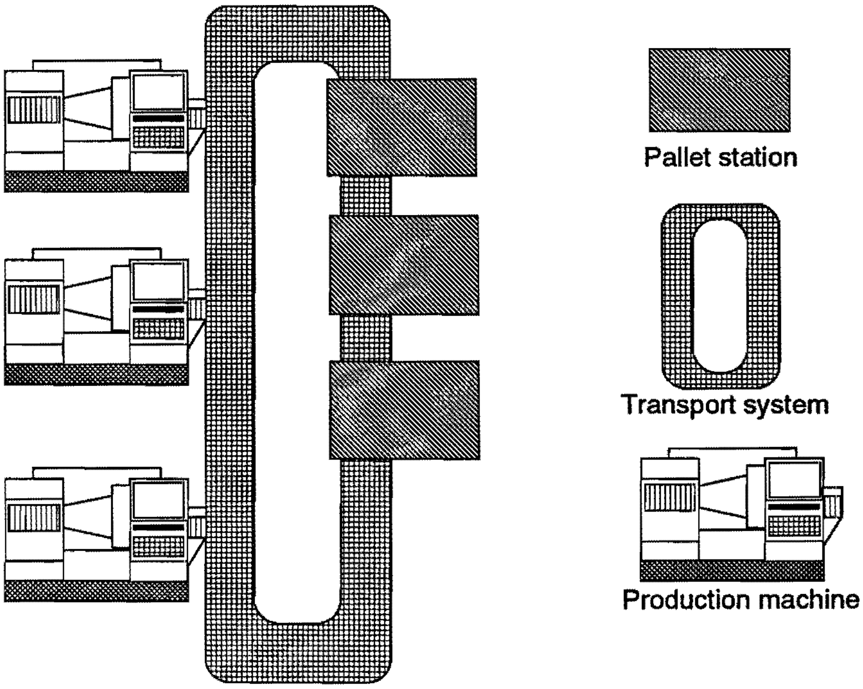


Figure 2: The Scharmann machine group

The system has been tested in the cost place 460 of the production department. This cost place manufactures the large parts of the turning machines, such as the machine beds. In this cost place two machine groups have been coupled to the monitor system. The first machine group exists out of one milling machine from the company Böhringer. This machine has two pallets. The machine uses currently the V24 protocol for the transfer of programs. For this machine a LSV2 protocol has been developed, which will be used for the program transfer and the data acquisition. The other machine group (figure 2) exists out of three milling machines from the company Scharmann and a pallet transport system. This machine group will be coupled to the monitor system with the V24 protocol for the program transfer and with digital signals for the data acquisition.

2. Analysis

The first step in the design of the monitor system has been the analysis of the information flow in and between the departments. In this analysis the structure of the environment of the flow, the objects in the flow, and the relations between the objects have been analyzed. This analysis is divided in five sections. These sections are the structure of the factory, the structure of the job generation, the definition of the objects in the production department, the necessary information on those objects, and the available information on those objects.

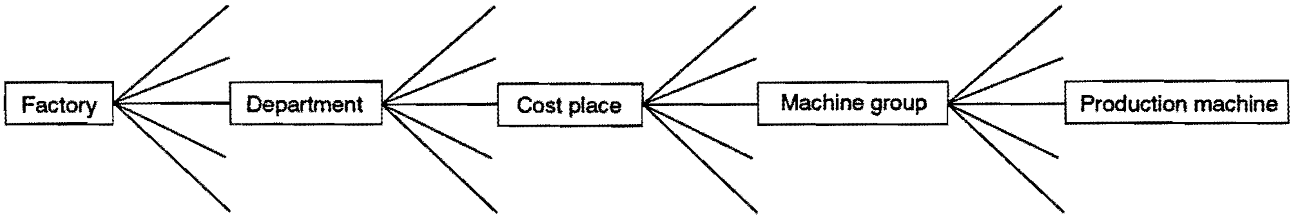


Figure 3: Levels in the factory structure

2.1. Factory structure

Inside the factory several structural levels can be observed (figure 3). The first level is comprised out of the departments. Each department has a special responsibility in the factory. In this report the following departments are important (figure 4):

- Planning department Plans the production and schedules the resources
- Cutting tool department Stores and distributes the cutting tools
- Auxiliary tool department Stores and distributes the auxiliary tools
- Programming department Develops the programs to fabricate the manufacturing parts
- Production department Executes the planned production with the required cutting tools, auxiliary tools, and programs

These departments are connected as seen in figure 4.

Each department exists out of several cost places, each having a special task within the responsibilities of the department. For example the production department is divided in cost

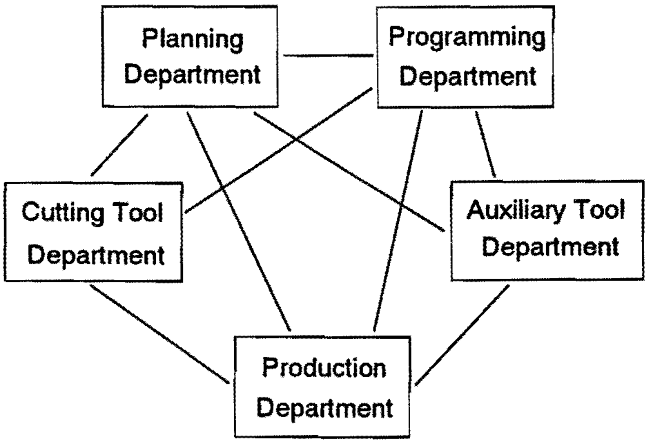


Figure 4: Information flows between departments

places according to the groups of manufactured parts, which are manufactured in the cost places. In this department cost place 460 is a production hall for the manufacturing of large parts.

In the production department the next level consists out of machine groups. A machine group contains one or more nearly identical production machines. The differences between the production machines are caused by differences in the reachable tolerances and in the composition of the tool storage magazines. The reachable tolerances of the production machines change as a consequence of the history of the machines. For example, a production machine normally used for rough milling, will not be able to reach the same tolerances as an identical machine only used for finish milling. The composition of the tool storage magazine is causing differences between the production machines because of the policy to change cutting tools as little as possible. In principle, the machines in a machine group are able to execute the same programs. The production machines form the lowest level in the factory structure.

The level of detail in the information flow between the departments varies as a consequence of the information handling of the different departments. The information transferred from the planning department to the production department is specified per machine group, while in the opposite direction the information is specified per production machine. Inside the departments the information has to be divided or combined to the necessary level of information.

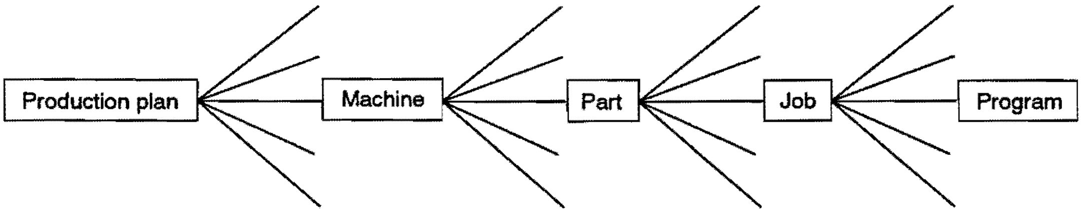


Figure 5: Levels in the job structure

2.2. Job structure

The structure containing the jobs is shown in figure 5. The basis of this structure is formed by the client orders and the production plan. The client orders and the production plan define the number and types of machines, that have to be manufactured before a certain date. The first step in the PPS-system is the division of the ordered machines in their separate manufacturing parts. For each of these manufacturing parts the manufacturing sequences are specified as a series of jobs. Sometimes a job can be executed in several machine groups. The choice between these machine groups is mostly based on the number of manufacturing parts, that has to be manufactured. One job is a series of manufacturing sequences, that have to be executed within a machine group. The sequences in a job can require several programs. In that case, for each program there exists a distinct fixture, which is used to position the workpiece during the manufacturing sequences in the program. The execution of a program can not be interrupted in order to start another job. When a manufacturing sequence uses multiple programs, it is allowed to manufacture the manufacturing part with several machines within the specified machine group.

Again the level of information in the information flow depends on the department. The information from the planning department is specified up to job and machine group level. In the opposite direction the information is available on program and production machine level. The monitor system has to be able to handle the differences in information level. This means splitting or combining the information into the correct level.

2.3. Object definition

The following section gives a description of the objects in the production process considered of importance to the development of the monitor system. The objects in the monitor system will be based on these objects.

2.3.1. Manufacturing part

A manufacturing part is a component in a machine. Within the company the process plans resulting in the manufacturing parts are available on the host computer. The process plan contains a short description of the manufacturing part and a list of manufacturing sequences resulting in the manufacturing part. Each manufacturing part is identified by a unique SNR-number (SNR = Sachnummer = Part number), which specifies the manufacturing part and all the manufacturing sequences of the manufacturing part. Each manufacturing sequence has its own AFO-number (AFO = Arbeitsfolge = Manufacturing sequence).

The manufacturing sequence contains a description of the sequence. A manufacturing sequence of a manufacturing part is always coupled to a specific machine group. The description lists the machine group number, the numbers of the required programs, and the auxiliary tools needed for the machining sequence. Furthermore, the description specifies the planned manufacturing times and setup times.

It is possible, that multiple programs are needed within one manufacturing sequence. In this case, each program has its own fixture. When a manufacturing sequence uses multiple programs, it is allowed to manufacture the manufacturing part with several machines within the specified machine group.

2.3.2. Master Job

A master job is a manufacturing sequence in a machine group on a batch of identical manufacturing parts. Within the company a batch of parts is identified by a unique ANR-number (ANR = Auftragsnummer = Order number). The number of manufacturing parts in the batch is specified in the master job description. Each master job corresponds with a

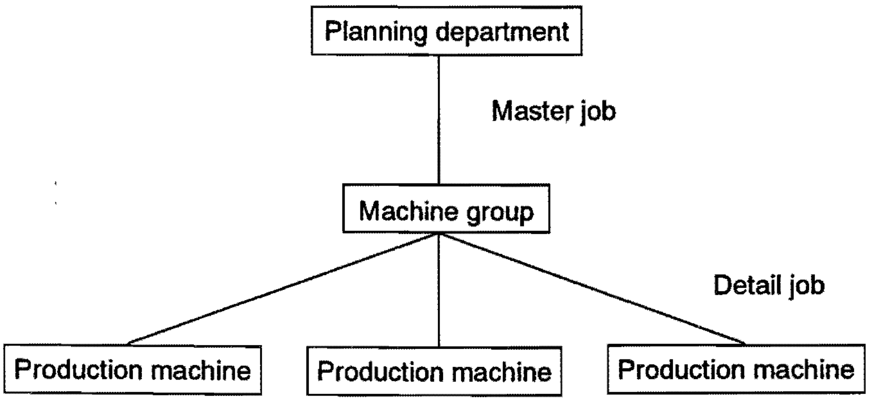


Figure 6: Difference between a master job and a detail job

manufacturing sequence of a manufacturing part. This sequence is defined by the SNR-number of the manufacturing part and the AFO-number of the manufacturing sequence. A specific master job is identified with its unique BNR-number (BNR = Belegnummer = Reservation number). This number is used to reserve cutting tools, production machines, and auxiliary tools. To influence the progress of a master job, each master job has a priority and a planned start- and end-time specified. If several master jobs are planned for the execution in a machine group within the same period, the priority of the job destines the succession of execution.

2.3.3. Detail job

A detail job is the manufacturing sequence of a batch of identical manufacturing parts on a specific production machine (figure 6). Each detail job corresponds to the repeated execution of one program. A detail job is defined by the BNR number of the corresponding master job and the program number executed within the detail job. The production machine is identified by its name. For each detail job the number of manufacturing parts is defined.

As a consequence of this definition, a master job consists out of several detail jobs. Only if one program is used in the master job and all the parts in the batch of manufacturing parts are manufactured on the same production machine, the master job will be equal to the detail job.

2.3.4. Program

A program contains all the instructions for the production machine to execute the manufacturing sequences of a manufacturing part in one fixture. Each program is developed for a specified machine group. The program is identified by a combination of the machine group number and a unique program number in the machine group. Within a program all the required cutting tools are defined. This list forms the base for the cutting tool difference lists.

2.3.5. Production machine

A production machine is a machine, which executes a program to manufacture a manufacturing part. Each production machine is identified by the cost place number, the machine group number, and a unique name. Each production machine has a tool storage magazine with cutting tools.

A production machine has a mode of operation. This mode of operation describes the production state of the production machine. Currently the following modes are considered:

- automatic: The production machine is usable for the production flow
- test-run: The production machine is unusable for the production flow because of a test of a program
- maintenance: The production machine is unusable for the production flow because of maintenance
- unknown: The mode of operation is unknown

To make it possible to specify more details to the mode of operation, the operating status of a machine has been defined. The operating status is one of the following: running program xx, failure, program xx interrupted, and setting up. If the operating status signals a failure, this failure is described by the alarm number, the alarm message, the source (LSV2-communication, NC-control or PLC-control) and the art of the alarm (fatal or warning). A fatal alarm stops the machine.

2.3.6. Pallet

A pallet is the supporting platform used for the transport of a manufacturing part in a machine group. Each pallet is defined by a unique number within the machine group. The use of a pallet is limited to a machine group.

2.3.7. Cutting tool

A cutting tool is the object used to remove material from a workpiece. The difference with the auxiliary tools is, that cutting tools have a limited life time due to the wear during the cutting sequences.

2.3.8. Auxiliary tools

An auxiliary tool is a tool needed in the manufacturing of a manufacturing part and not one of the other resources. Examples of auxiliary tools are fixtures or measuring tools. The life time of an auxiliary tool is not important. At the moment a more detailed specification of an auxiliary tool is impossible, due to the development of a new tool handling method.

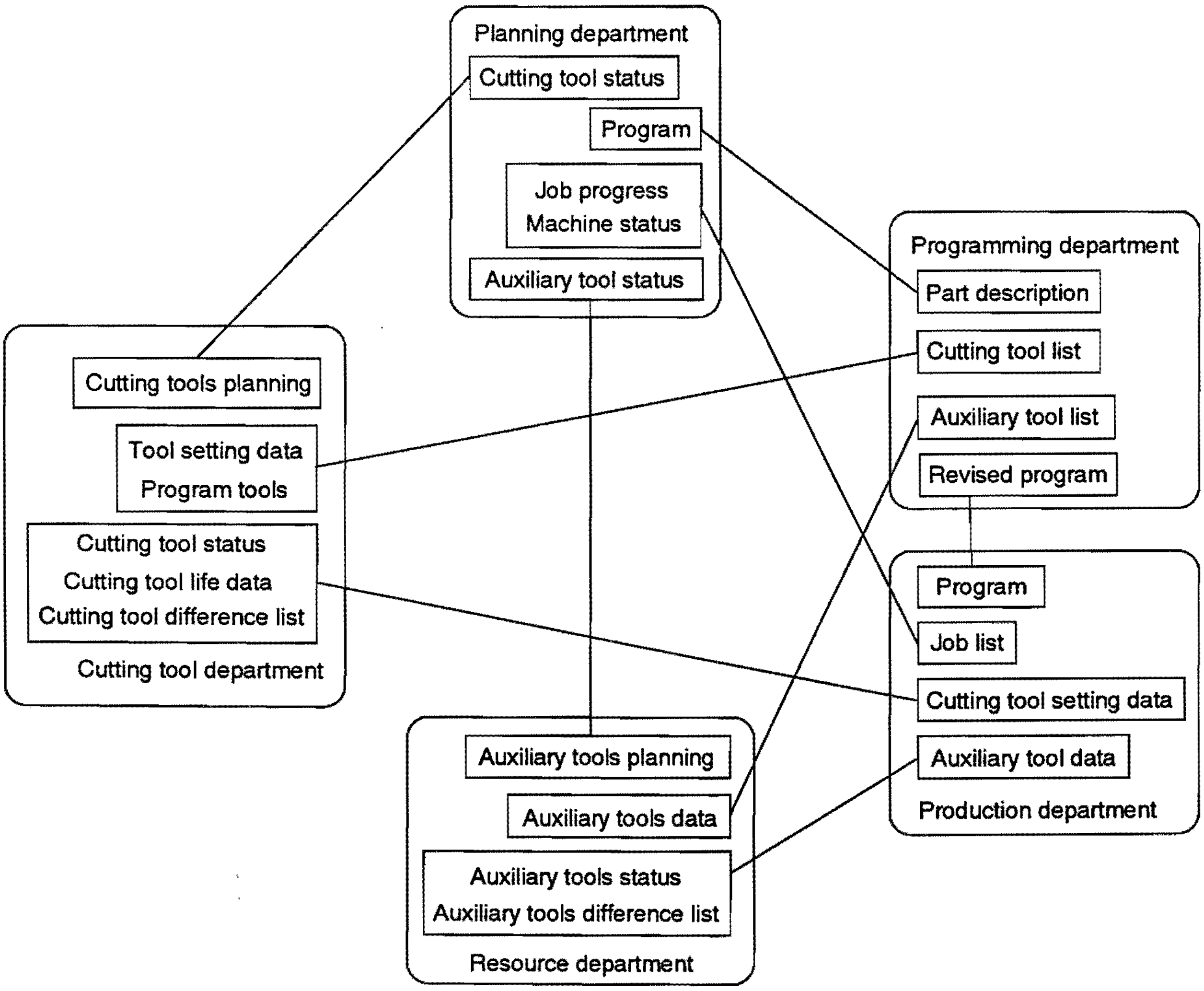


Figure 7: Necessary information in the departments

2.4. Necessary object information

Each department has its own requirements to the information connected to the objects. In the following section a description is given of these requirements specified per department.

2.4.1. Planning department

The planning department generates the master job lists for the production department starting from the year-production plan. To realize this, the planning system needs lists of data, defining the decomposition of the ordered machines into the manufacturing parts. At the moment, when the lists of manufacturing parts have been generated, the planning system needs the process plans defining the manufacturing sequences, which have to be executed to create the manufacturing part. For each manufacturing sequence has to be defined the machine group, the auxiliary tools, the programs, and the cutting tools needed for the manufacturing sequence, together with the required manufacturing times and setup times. With this information the system can create a master job list.

While generating the master job list, the planning department only checks the availability of the workpiece material. In order to make a more detailed planning of the jobs, the planning department needs the state of all cutting tools, auxiliary tools, production machines, and jobs. These states must define the current job using the resources, the remaining tool life of the cutting tools, and the availability of the resources.

The output of the planning department is a cutting tool planning, defining the use of the tools for the cutting tool department, an auxiliary tool planning, defining the use of the auxiliary tools for the auxiliary tool department, and a master job list, describing the job and necessary resources for the production department.

2.4.2. Cutting tool department

The cutting tool department handles the distribution and the storage of the cutting tools. In order to distribute the tools, the department needs a schedule, defining the planned use of the cutting tools. In this schedule has to be listed the number and types of the tools, the required tool life of the tools, the machine group requiring the tools, and the dates and times the tools are needed. This schedule can originate from the planning department or, in the form of cutting tool difference lists, from the production department. To make it possible to preset the tools, the required presetting data has to be defined.

When the programming department develops a new program requiring non-existing cutting tools, the cutting department has to be notified. Subsequently this department can order new tools.

To keep track of the states of the tools, the tool department needs reports specifying for each tool the used tool life and major occurrences to the tools, such as the break of a tool or a collision between a tool and a workpiece.

The output of the cutting tool department, is a list of all existing tools for the programming department, a list of tools and their states for the planning department, and a tool presetting list defining the tools and data for the production department.

2.4.3. Auxiliary tool department

The auxiliary tool department handles the distribution and the storage of the auxiliary tools. In order to distribute the tools, the department needs a schedule, defining the planned use of the auxiliary tools. In this schedule has to be listed the number and types of the tools, the destination of the tools, and the dates and times the tools are needed.

When the programming department develops a new program requiring non-existing auxiliary tools, the auxiliary department has to be notified. Subsequently this department can order new tools.

The output of the auxiliary tool department, is a list of all existing auxiliary tools for the programming department, a list of auxiliary tools and their status for the planning department, and a list defining the auxiliary tools and related data for the production department.

2.4.4. Programming department

In order to develop a program for a new manufacturing part, the programming department requires a description of the part. While developing a program, it can be useful to have access to lists defining all existing cutting tools and auxiliary tools.

The programs developed by the programming department uses machine independent programming languages. To be usable in a specific machine group, this program has to be translated with the help of a post processor. This proceeding makes it easy to adapt a program to different types of machine controls. To keep track of changes in the programs, the programming has to be notified in the case of revisions of a program.

The output of the programming department is a program for the production department, a list of required tools for the cutting tool department and auxiliary department, and the manufacturing data for the planning department.

2.4.5. Production department

To be able to produce the required workpieces, the production department needs a master job list specifying the jobs. In this list the necessary cutting tools, programs, and auxiliary tools have to be defined, together with the planned manufacturing period. To be able to use the cutting tools, the controls of the production machines require setting data and tool life data of the cutting tools, and the required position of the tool in the tool storage magazine.

The output of the production department is the status of the jobs and production machines for the planning department, the status of cutting tools for the cutting tool department, the status of the auxiliary tools for the auxiliary tool department, and revised programs for the programming department.

2.5. Available object information

Although much of the required information is available there are still several problem areas. In the planning department all data is available to generate master job lists. However, due to the absence of information on the states of any resource, the planning department is not able to generate a detailed planning. The monitor system has to improve this situation.

At the moment little information is available, concerning the cutting tools and the auxiliary tools. This is due to the development of new tool handling methods. Although the required cutting tools are listed in the programs, it is difficult to generate cutting tool difference lists. The major problem is the absence of a detailed scheduling of the jobs to the production machines. Therefore it is impossible to decide, which tools are really needed on a production machine.

The same holds for the auxiliary tools. In this case the problem is intensified, while the specification of the required auxiliary tools is listed per machining sequence, and not per program. Therefore, it is possible, that a job waits for an auxiliary tool only needed in one of the program, while it is possible to execute one of the other programs.

A problem with the information originating in the programming department is the absence of any required tool life data in the programs. To generate an effective planning of the cutting tools, it is imperative to incorporate this information in the programs.

Almost all the data originating in the production department, is connected to the production machines. In order to access this data, a transmission protocol has to be used. In this case, the majority of the data transfers is executed with the help of the LSV2 data transfer protocol. The available data and the format of the data are dependent on the controls of the production machine. A survey of the availability of some important data is found in the next table. In this table the availability of the data is listed for the TRAUB TX8D control, the Heller control and the Sinumerik 850 control. These controls are used by the different production machine types within the production department of TRAUB. For each data object is listed, if it is available directly (colon D), if it is possible to get the information indirectly (colon I), if the data can be requested from the production machine control (colon R) and if it will be send automatically (colon A). The codes in the table are:

•	TRAUB TX8D				Heller				Sinumerik 850			
	D	I	R	A	D	I	R	A	D	I	R	A
program number	TA		TC	TD	+		+	+	SF		SF	
workpiece counter	TB		TC	TD		HE			SA			SA
machining start	TE		TC	TD	HB		HB	HA	SA			SA
machining end	TE		TC	TD	HB		HB	HA	SA			SA
run-time workpiece	+		TC	TD		HE				SB		
run-time job	TG		TC	TD		HE			SA			SA
automatic mode	+		TC	TD	+		+	HA	+		+	
test run mode	TF		TC	TD	HC		+	HA				
maintenance mode	+		TC	TD	HC		+	HA				
program running	+		TC	TD	+		+	HA		SE		
failure		TH			+		+	HA		SE		
setting up	+		TC	TD	HC		+	HA				
alarm number	+		TC	TD	+		+	HA	+			+
alarm source NC						HD			SC			SC
alarm source PLC						HD			SC			SC
alarm source LSV2	+			+	+			+	SC			SC
alarm type fatal	+		TC	TD	+		+	HA	+			+
alarm type warning	+		TC	TD	+		+	HA	+			+

table 1: Available information for LSV2 communication with specific control systems

+ data available
empty data not available
TA available if program started by a computer
TB available if workpiece counter is set by a computer
TC not available in automatic mode
TD available in automatic mode with the correct status mask
TE signal is used for more than one goal
TF data has to be changed by operator
TG can be calculated from run-time workpiece and workpiece counter
TH available in the alarm reports
HA available with the correct status mask
HB signal is used for more than one goal
HC different names of modes
HD is available from the interval of the alarm number
HE can be calculated from machining start and machining end signals
SA only available for a pallet and its job
SB can be calculated from the workpiece counter and the run-time of the job
SC alarm reports have different formats
SD data has to be changed by operator
SE is available in the alarm reports
SF only possible with a synchronization procedure for all pallets and complete tool storage magazine

The differences in the availability and the format of the data can cause problems in the development of the monitor system.

3. Framework

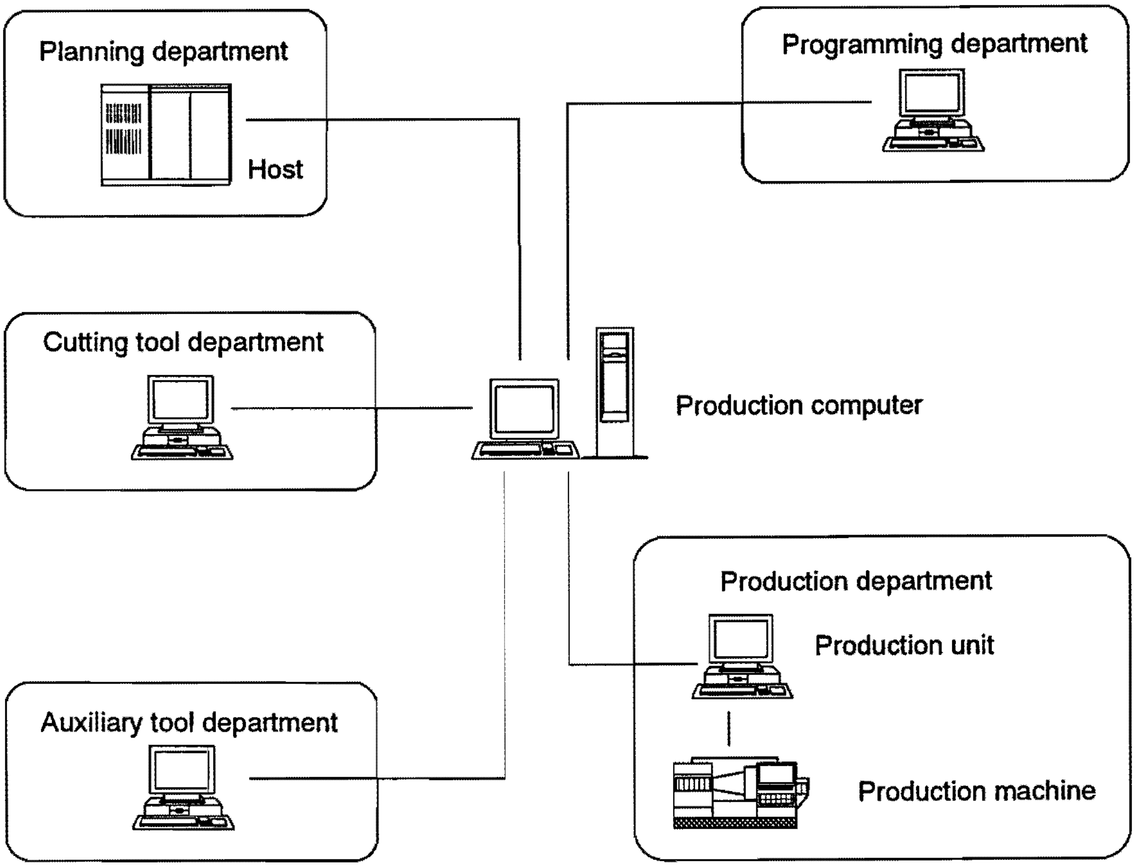


Figure 8: Hardware configuration of the monitor system

3.1. Hardware configuration

The monitor system exists out of several hardware systems (figure 8). The most important system is the production computer. This IBM-AIX 6000 computer is the central computer in the monitor system. The AIX operating system of the production computer is a dialect of the UNIX multi tasking standard. All communications in the monitor system are coordinated by this computer.

The IBM-mainframe host is the central database-system within the company. All information is stored on this system. The PPS-system generating the master job lists executes on this computer. Other systems have access to the information in the database using an ADABAS SQL-interface.

The CAD-stations in the programming department are only loosely coupled to the monitor system. If a new program has been developed, the programming department downloads the programs to the production computer. There the other departments can access these programs

The production units serve as interface between a production machine, the production machine operator and the production computer. The production unit is an 80286 AT compatible computer. The XENIX operating system is a dialect of UNIX developed by MICROSOFT. These units are used for the data acquisition concerning the progress of the jobs and the states of the resources in the production department. The machine operator must be able to review all data relevant to the machine group and its jobs, and to execute transfers of programs to and from the production machine and the production computer. To make the access to the monitor system as easy as possible, there may be several production units in each machine group (figure 9).

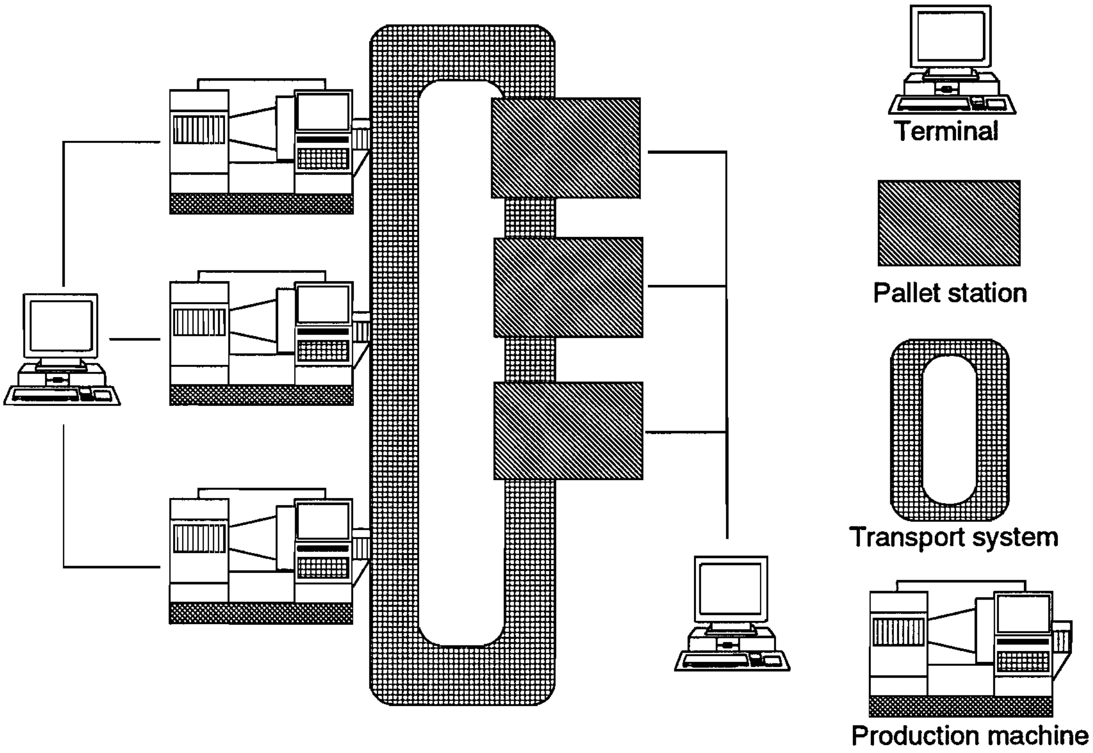


Figure 9: The production units in the Scharmann machine group

At the moment, the implementation of the monitor system is limited to the production department. The cutting tool department and the auxiliary tool department will be incorporated in a later stage. It is to be expected, that the hardware and software will be comparable to the implemented hardware and software of the production units.

3.2. Data objects

For the implementation of the monitor system all information has been grouped in a set of data objects. Each data object contains data specific to an object in the production department and planning department. It is tried to reduce the double information to a minimum. This resulted in data objects, which contain references to other data objects. This means, that it may be necessary to get information from several other data objects to display all the relevant information of a single data object. This information can be distributed over several software modules.

The following data objects are defined:

- Master-job

  This is a description of a job in the master job list. This object contains the following data-fields:

- BNR: Identification of the job in the master job list
- SNR: Identification of the manufacturing part, that has to be manufactured
- AFO: Identification of the manufacturing sequence, that has to be performed
- planned: Number of manufacturing parts, that have to be manufactured in this job
- EP: Priority of the job
- A_Zeit: Planned start of manufacturing period. (Anfang Zeit)
- E_Zeit: Planned end of manufacturing period. (Ende Zeit)

- Detail job

  This is a description of a job in the detail job list and its progress. Each detail job is restricted to one production machine. This object contains:

  - ident: Identification of specific production machine, which executes the job
  - BNR: Identification of the master job
  - program: Number of the program, that has to be executed
  - produced: Number of finished manufacturing parts
  - planned: Number of planned manufacturing parts
  - state: State of the job
  - B_Zeit: The actual total manufacturing time. (Beleg Zeit)
  - L_Zeit: The actual total time since the start of the job. (Lauf Zeit)

- Manufacturing part

  This object contains the description of all the manufacturing and assembly sequences needed for the manufacturing of one manufacturing part:

  - SNR: Identification of the manufacturing part
  - description: Short description of the manufacturing part
  - sequences: List of manufacturing and assembly sequences. For each sequence there exists:
    - KST: cost budget number of production department
    - MGR: identification of machine group, in which the sequence has to be executed
    - programs: List of programs needed for this sequence
    - resources: List of auxiliary tools needed for this sequence
    - actions: Description of details of the sequence
    - ST_Zeit: Planned manufacturing or assembly time. (Stück Zeit)
    - R_Zeit: Planned setup time. (Rüstung Zeit)

- Program

  This object exists out of an actual program and relevant data:

    - ident: Identification of the program. This is a combination of the machine group number and the program number
    - number: The program number
    - program: The actual program
    - tools: List of required cutting tools

- Production machine

  Description of the state of a specific production machine. This data object has the following fields:

  - ident: Identification of production machine in the machine group
  - KST: Identification of cost budget of the production department
  - MGR: Identification of the machine group
  - mode of operation: Description of the current mode of operation
  - operating status: Description of the current operating status
  - alarms: Description of existing alarms

  - programs: List of programs available in the memory
  - tool magazine: List of available cutting tools in tool storage magazine
- Pallet

  Description of the state of a pallet:
  - ident: Identification of pallet within pallet carrousel
  - BNR: Identification of job
- Cutting tool

  Description of a cutting tool
- Auxiliary tool

  Description of an auxiliary tool

4. Implementation

The implementation of the monitor system is limited to the host, production computer, and production units. Due to the development of new handling methods, currently the information concerning the cutting tools and the auxiliary tools is not monitored. The basis of the monitor system is the FREENET 10000 software package developed by the German company RWT. During this project I have developed several new processes in order to implement the required functionality of the monitor system. These processes are the LSV2 processes for the communication between the production units and the production machines, and the monitor process, which generates the overviews of the states of the production machines and of the progress of the jobs. All other processes are commercially available and have been adapted to this monitor system by changing the configuration files of these processes. In the following section a description is given of the processes used by the monitor system.

4.1. Host

At this moment there is no software available, which makes direct access to the host database possible. Therefore a partial solution is implemented. In this solution the planning department has developed a process, which downloads the master job list automatically to the production computer after each planning run of the PPS-system. The downloaded file automatically replaces the file containing the old master job list. In this file the master job list is represented as a list of programs, listing per program the SNR number, AFO number, and the name of the manufacturing part, the ANR number of the order, the number of manufacturing parts, the planned start date and end date of the job, the BNR number of the job, the program number and the program suffix. Because of the development of new handling methods, no data is downloaded concerning the auxiliary tools and cutting tools.

The data gathered by the monitor system is stored in files on the production computer. As a consequence, it is only possible to create an overview of the system on the production computer. Because of current limitations in the available software, this data can not be accessed by the planning department or any other department.

4.2. Production computer

The processes of the production computer are implemented with the help of the FREENET 10000 software package developed by the German company RWT. To this package a monitor process has been developed, which combines the data in the system and generates log files storing the data concerning the progress of the jobs and the state of the machines.

4.2.1. FREENET processes

The FREENET processes on the production computer exist out of five processes (figure 10). The first process is the network handler. This process executes the communication between the FREENET processes on the production computer and the FREENET processes on the production units. The communication between the computers uses an Ethernet network. The communication between the computers is executed with the TCP/IP protocol.

The second process is the network coordinator. This process coordinates the access of the FREENET processes to the network. At each moment, only one process has access to the

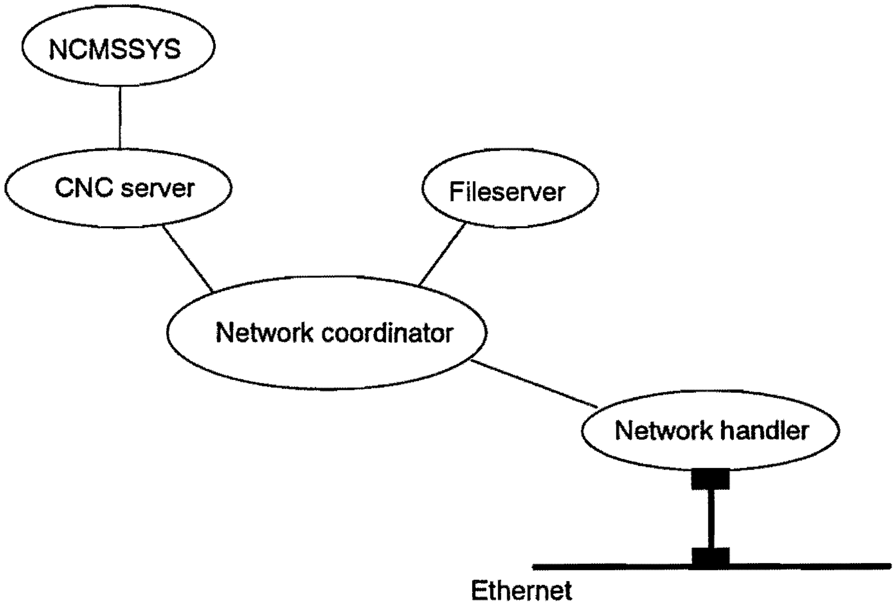


Figure 10: FREENET processes on production computer

network.

The third process is the file server. This process handles the transfer of files between the production computer and the production units. The commands to transfer a file are generated by the NCMSSYS process on the production computer and the terminal processes on the production units. The file server transfers two sorts of files: the FREENET configuration files and the program and difference list files of the monitor system. The configuration files for all the processes on the production units are stored on the production computer, to simplify the maintenance of the system. Each time one of the production units starts, the configuration files are downloaded to the unit.

The fourth process is the CNC server process. The CNC server coordinates the distribution of the messages between the processes. Inside the FREENET software several groups of messages exist, each having a special limited scope. Examples of groups of messages are the group containing the commands inside FREENET, the group containing data messages, and the group reporting the start of processes. At the start of each process, the process has to connect to the CNC server and notify the server process, which messages it wants to receive. During normal execution each process wanting to send a message, sends the message to the CNC server, which in turn distributes the message to all processes, requesting the reception of the group of messages. The messages are implemented with the help of the messages defined in the UNIX system 5 operating system standard.

The fifth process is the NCMSSYS process. This process handles the startup of the production units and the execution of the FREENET commands. When this process is notified of the start of a production unit, it reads the configuration files defining the monitor system and, based on this information, orders the file server to download the necessary configuration files to the production unit. When a command is received, the process starts the execution of the necessary procedures and at the conclusion notifies the CNC server of the results of the command. The CNC server reports the results to the processes triggering the command.

In a later stage a sixth process will be connected to the CNC server. This process will automatically execute the download of the information in the master job list and the necessary programs to the production units, when a new master job list is received. Currently this process is not yet available for the IBM AIX 6000 production computer.

4.2.2. Monitor process

The monitor process is developed to generate overviews of the states of the production machines and of the progress of the jobs with the help of the messages received from the FREENET software. This process is connected to the CNC server process on the production computer. The process logs all changes in files. Per machine the process uses one file for changes to the machine status and one file with the changes to the job status.

The process uses the STARTGROUP and MDEBDE messages to generate the overviews. The STARTGROUP messages notify the process of starts of production units. Each time a new production unit starts, the FREENET software on the production computer receives a STARTGROUP message. This message contains the name of the starting production unit. This message implicitly indicates, that no software has been running on the production unit. This means, that the production unit has been unable to report changes in the states of the production machines connected to this unit. Therefore, all information in the monitor process concerning this production unit is invalidated. When the monitor process receives the STARTGROUP message, it reads the configuration file defining the production machines connected to this system and resets the variables tracking these machines.

The MDEBDE messages report changes in the states of the resources and jobs. Each message contains the name of the production machine triggering the message, the time and date of creation, and the data specifying the change. The MDEBDE messages can be divided in two groups: the messages reporting changes of program or pallet numbers, and the messages reporting changes to the states of objects.

If the process receives a message reporting the change of a program number, the process calculates the number as a first step. If the complete number is defined in the message, this is simply reading the number in the message. If the message reports the change of a bit in the program number, the process calculates the number by changing the bit in the program number. The process keeps track of all the bits in a program number to decide, if the value of the bit is known. The values of all the bits in a number are known, if the complete number is changed in one time or if all the bits in the number have changed at least once. If the values of all the bits are known, the process tries to find the job number of the job running on the production machine by reading the master job list and taking the first job number, which has the program listed in its specifications. At the moment the process logs all changes of numbers.

If the process receives an other MDEBDE message, the machine logs the change in the corresponding files. If the change is the start, end, or interruption of a program, the process logs the date of time of the change, the type of the change, the number of the corresponding program, and, in the case of the end and interruption of a program, the time elapsed since the last change in the program status. If the number of a job is known, the same information is logged for the corresponding job. If the change is the start or end of a mode of operation, an error, or a warning, the process logs the date of time of the change, the type of the change, and the time elapsed since the last change of the corresponding detail.

The process uses a special file to update the data in the process. With this file it is possible to define the states of the production machines, if the information concerning the production machine has been invalidated by the crashing of a production unit. Each time this file is created, the process reads the file, changes the states of the specified production machines, and deletes the file. The process only changes those details, which are unknown at the moment of reading.

4.3. Production machine unit

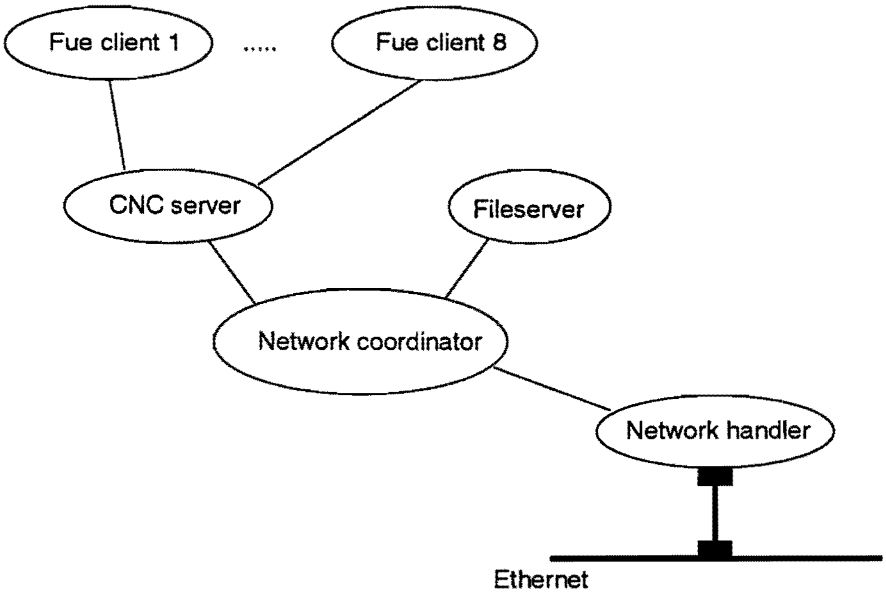


Figure 11: FREENET processes on a production unit

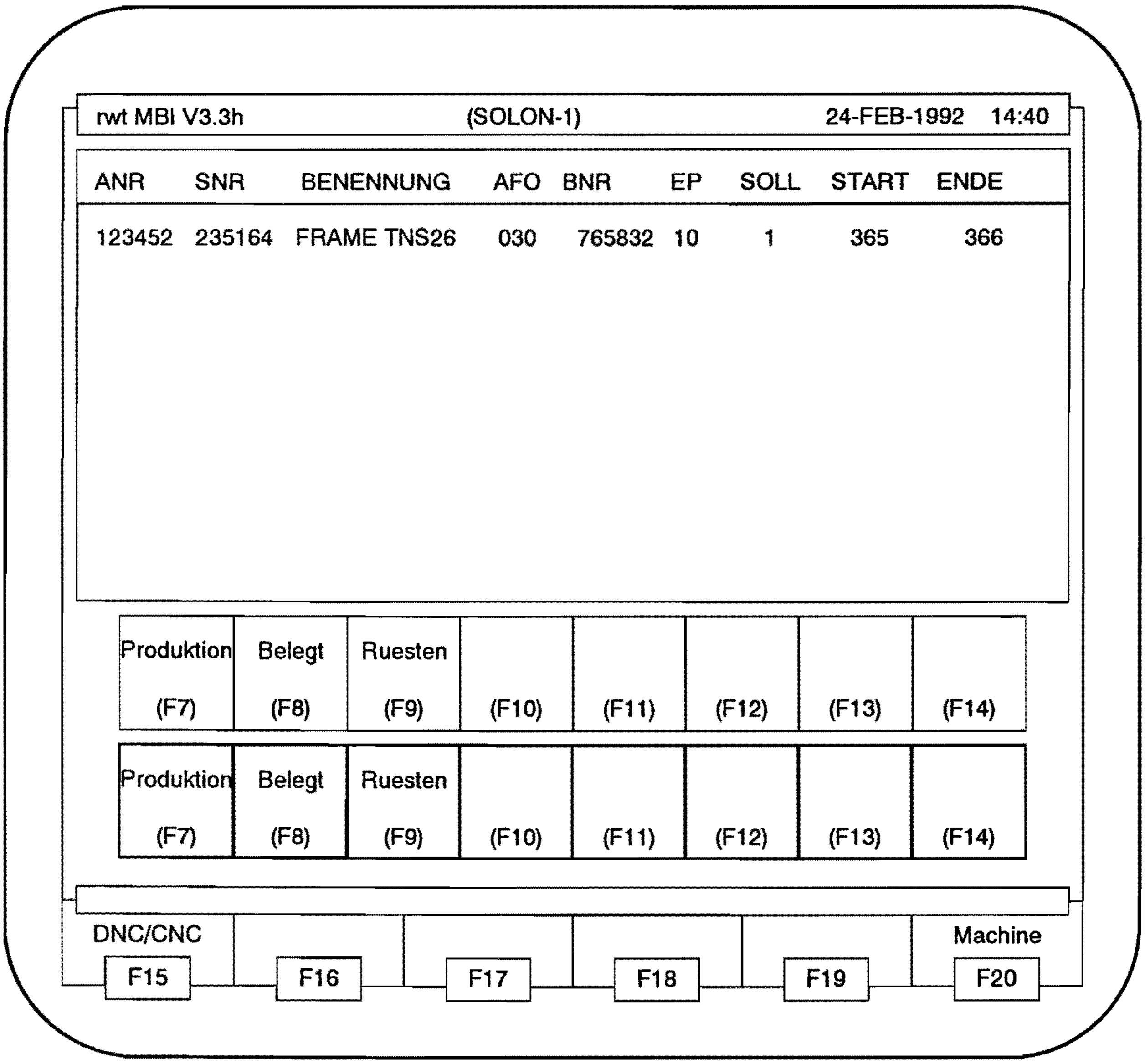
4.3.1. FREENET processes

The FREENET processes on a production unit exist out of five groups of processes (figure 11). Four of those processes are the counterparts of the identical named processes on the production computer. These processes (the network handler, network coordinator, file server, and the CNC server) have the same responsibilities as their counterparts on the production computer.

The fifth group constitutes out of the so-called Fue Client processes. Each production machine has two client processes. The first process is responsible for the data on the terminal display and in the database concerning the production machine. The other process controls the communication with one of the external device drivers.

4.3.2. Database

Each production unit contains one database. The database used by the FREENET processes is the commercially available package INFORMIX SQL. In the current situation, only the FREENET processes have access to the database. The database is used for the storage of the states of production machines, of programs, and of messages. The data in the database is controlled by the Fue Client processes. Each machine has one client process, which has access to the database to change values.



rwt MBI V3.3h (SOLON-1) 24-FEB-1992 14:40

ANR	SNR	BENENNUNG	AFO	BNR	EP	SOLL	START	ENDE
123452	235164	FRAME TNS26	030	765832	10	1	365	366

Produktion (F7)	Belegt (F8)	Ruesten (F9)	(F10)	(F11)	(F12)	(F13)	(F14)
Produktion (F7)	Belegt (F8)	Ruesten (F9)	(F10)	(F11)	(F12)	(F13)	(F14)

DNC/CNC F15	F16	F17	F18	F19	Machine F20

Figure 12: Mask of the terminal display with the function keys

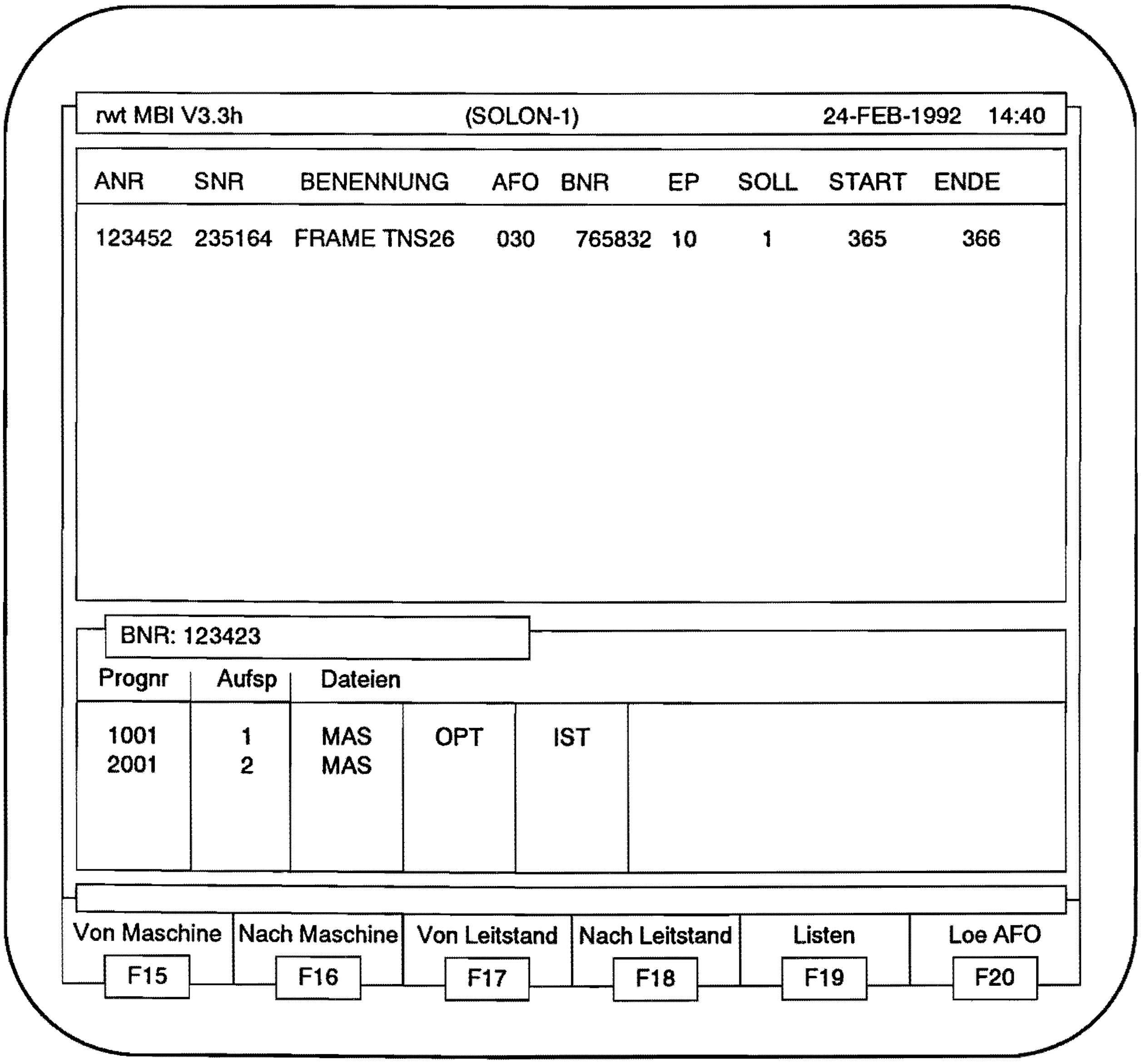
4.3.3. Terminal handler

The displaying of data on the terminals is handled by the terminal handlers. The handlers do not have direct access to the data, but are connected to a Fue Client process. Each terminal has one terminal handler. If a terminal is used to control several production machines, the terminal switches between the client processes of the production machines to get the necessary data. The function of the terminals are: displaying of the jobs planned for a production machine, displaying the state of the production machine, and triggering the transfer of programs between the production unit and the production machine, and the production unit and the production computer.

The behavior of the terminal can be configured with the help of a programming language developed by RWT. The programming language is executed by the Fue Client process connected to the terminal handler. In this programming language, the display of the terminal is a group of windows. With the help of the programming language, it is possible to define, which field of the database is to be displayed. The programming language has possibilities to set the values of the field of the database, but has no possibilities to execute calculations with those fields.

In order to monitor the state of a production machine sixteen Boolean status variables are available. The value of these variables can be changed with the help of function keys and signals. The function keys correspond to the F7-F14 function keys. In the programming language it is possible to define actions, which are triggered by the function keys and the signals. The results of these actions are messages send to other processes and changes of the status variables.

Two configurations of the display are used. The first (figure 12) shows a window containing the list of jobs scheduled for this machine group and a window with the function keys. The current status of the status variables is shown by the highlighting of function keys. The second configuration (figure 13) shows the same window with the list of jobs and a second window containing a list of files. These files are the files connected to the job, which is selected in the first window. When needed, the terminal handler switches automatically between the



rwt MBI V3.3h (SOLON-1) 24-FEB-1992 14:40

ANR	SNR	BENENNUNG	AFO	BNR	EP	SOLL	START	ENDE
123452	235164	FRAME TNS26	030	765832	10	1	365	366

BNR: 123423

Prognr	Aufsp	Dateien			
1001 2001	1 2	MAS MAS	OPT	IST	

Figure 13: Mask of the terminal display with the program lists

configurations.

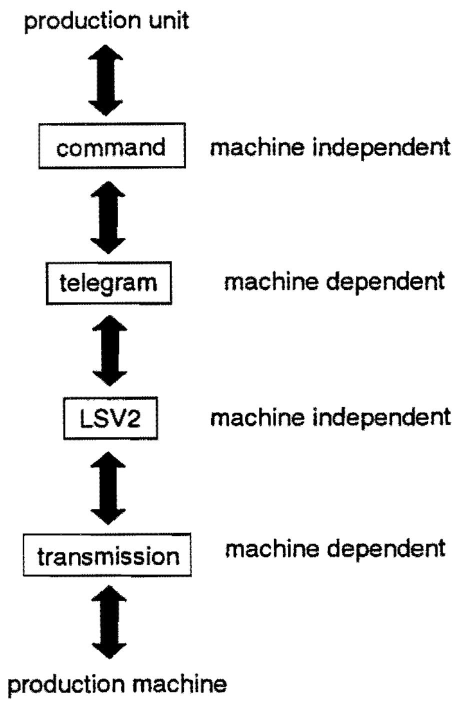


Figure 14: Communication levels between a production unit and a production machine

4.3.4. LSV2 processes

Wherever possible, the communication between the production unit and the production machine will be executed with the help of the LSV2 data transmission protocol. This protocol standardizes the communication between two a-synchron functioning stations. The controls of the production machines, which can communicate with the help of this protocol, have in general extended facilities for automatic data acquisition. Because the FREENET software does not have LSV2 capabilities, two LSV2 implementations have been developed: one for the Sinumerik 850 LSV2 protocol, one for the TRAUB TX8D LSV2 protocol. Both controls are used in the monitor system.

The communication between a production unit and a production machine can be divided in to four levels (figure 14). The first level constitutes out of the commands from the monitor system and the messages to the monitor system. An example of a command, is the command to transfer a file to the production machine. An example of a message is the message to the monitor system containing the state of a production machine. The format of the commands and messages will be machine independent.

The next level constitutes out of the machine telegrams. The machine telegrams contain the data, which is transmitted between the production unit and the production machine. The machine telegrams are machine dependent. As a result of differences in the format and contents of telegrams, one telegram can result in several messages, or one command will need several telegrams. In the case of the command to transfer a file to the production machine, the command is split into multiple telegrams. First a telegram has to be send to the production machine to initiate the transfer. After a positive answer from the machine, several telegrams, each with a section of the program, have to be sent to the machine. At the end of the transfer a telegram has to be send, reporting the end of the transfer.

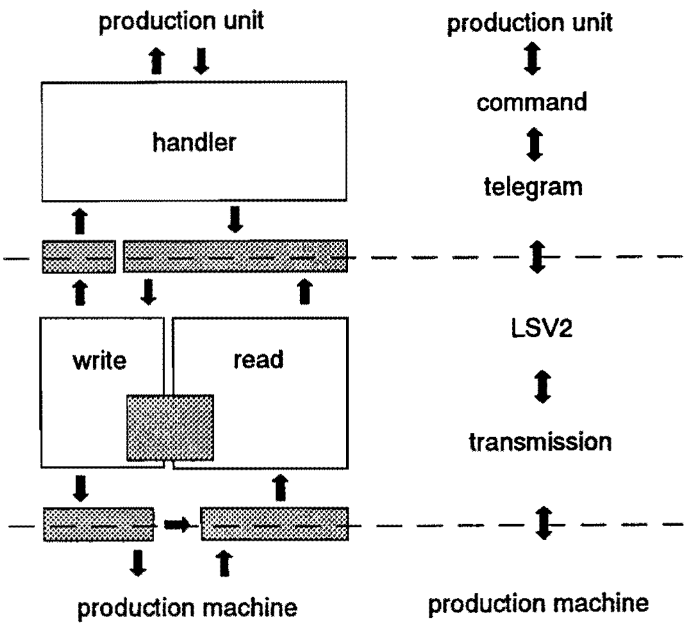


Figure 15: Process structure for LSV2 communication

The next level is the LSV2 protocol. This is the communication interface between server and production machine. This protocol defines the way to coordinate the transfer of a machine telegram between two a-synchron stations. At this level the sending station will start the communication. At a positive acknowledgment from the receiving station, the telegram is sent including a checksum. The receiving station will report the sending station, if this checksum is in order. In this case the sending station will end the communication. The LSV2 protocol is supposed to be machine independent. Because several control developers have implemented dialects of the protocol, the actual protocol can differ between machines.

The last level is the level of the port controls. In this level the baud rate and parity of the communication are described and the actual transfer of characters takes place. Due to the differences in the capabilities of the machine controls, this level is machine dependent.

The machine interface module exists out of three processes running parallel to each other (figure 15). These processes are the handler process, the read process, and the write process. The handler process controls the information flow between the monitor system and the production machine. For the transfer of the data the handler process is dependent on the read and write processes. According to the received commands and telegrams, the handler process sends the correct answers to the sender and transformed data to the receiver. In the handler process the transformation from commands to telegrams will be executed.

The read process reads the characters send from the production machine. According to the characters, it triggers actions in the write process. Because the production machine has a priority in the transmission, the read process will have a priority to the write process.

The write process executes the communication protocol with the production machine. It establishes the connection, transmits the telegram and closes the connection. This process controls the transfer of a telegram, but is depended on the read process for the reactions of the production machine. The combination of the write process and the read process controls the LSV2 level and the transmission level of the communication.

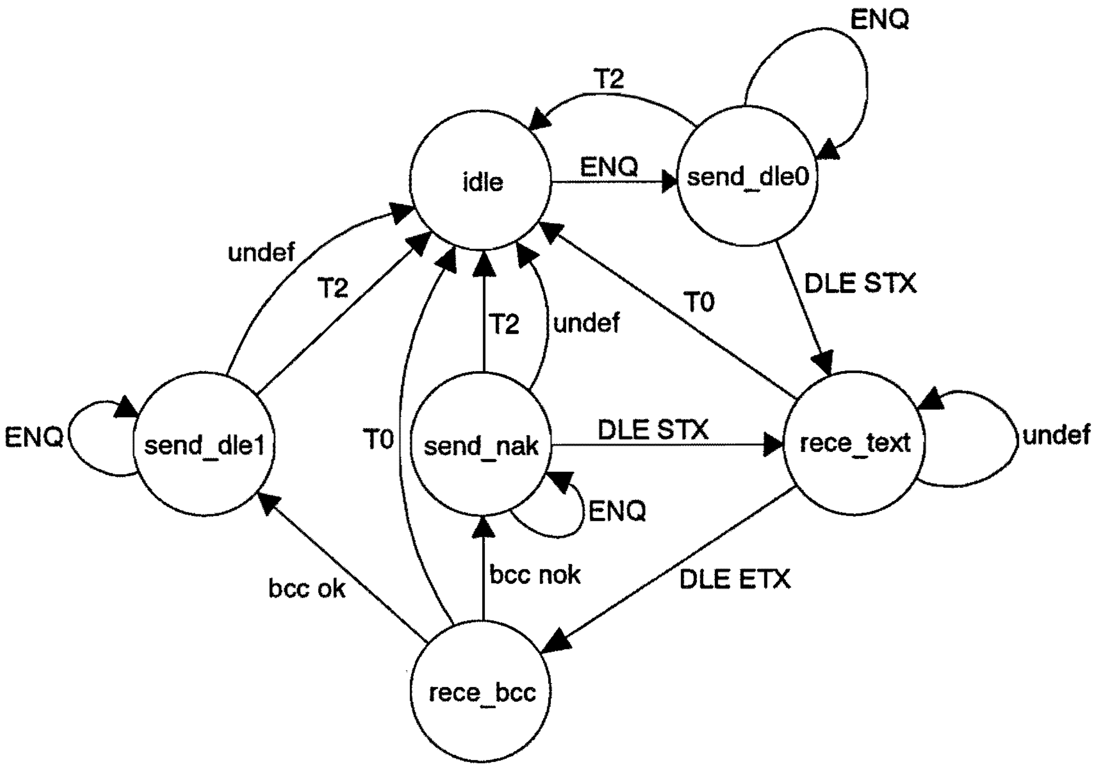


Figure 16: State flow diagram for read process LSV2 communication

These processes are programmed as finite-state machines. This means, that each process has a set of possible states (figure 16 and 17). In each state it can receive a set of signals, which transforms the state to another state. Each state corresponds to a set of actions, which have to be executed. After the execution of these actions, the process waits for new signals.

The programs are written with the help of a so-called WART preprocessor. The preprocessor translates special formatted files into normal files for the C programming language. This preprocessor is used, because the specially formatted files are better readable as the resulting C files. The files consist of two sections. The first section is the procedure, which handles the finite-state machine. This section contains groups of statements, which are executed when in a defined state a token is received. The first statement in the group defines the state and token, which triggers the execution of the following statements. The last statement in the group defines the new state of the finite-state machine, which is entered after the execution of the statements in the group. The other section contains the procedures in normal C language, which are called by the finite-state machine.

The read process and the write process are synchronized with the help of shared memory and semaphores defined in the UNIX system 5 standard. In the shared memory a semaphore is declared, which both processes can access. However only one process at a time can own the semaphore. Most of the time, the read process will own the semaphore. If a read of the port does not result in the reception of a character, the read process will release the semaphore and immediately claim the semaphore again, in order to try an other read of the port. This sequence will be repeated, until a character has been read from the port. The write process only tries to claim the semaphore, if the process wants to write a character to the port. If the process is successful in claiming the semaphore, it checks if the read process has send a message to the write process. If this is not the case, the write process writes the character to the port, changes the state of the read process, and clears the port buffer of the read process.

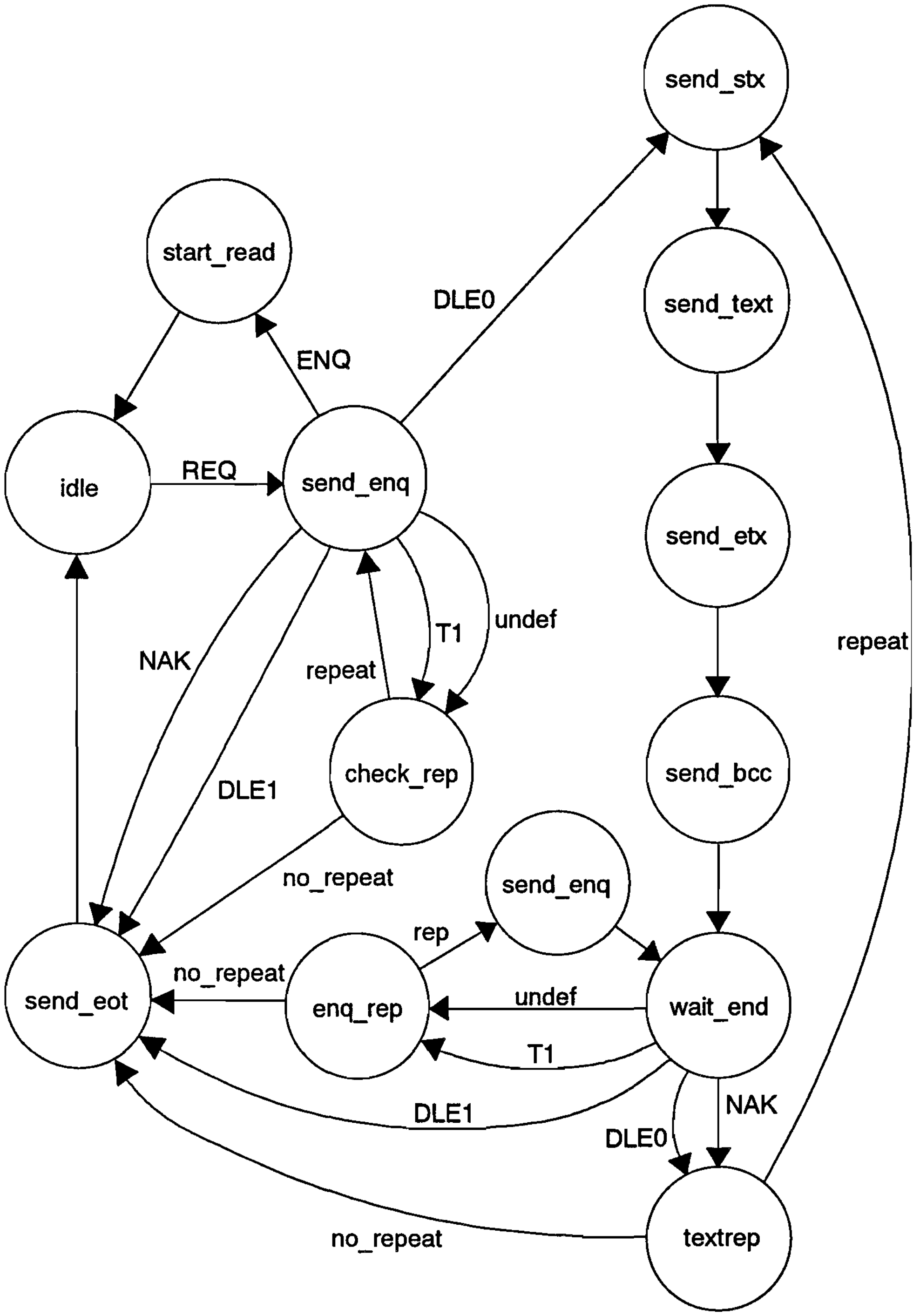


Figure 17: State flow diagram for write process LSV2 communication

Then it releases the semaphore. If the read process has sent a message, the write process does not send the character, but releases the semaphore. Depending on the message, the write process notifies the handler process of the failure to transmit, or tries to re-transmit the character.

The communication between the processes is handled with the messages defined by the UNIX system 5 standard. If the write process receives a message from both of the other processes, the messages of the read process have priority. If the write process receives a message from the handler process, while engaged in another transmission, the write process rejects the message, and the handler has to retry in a later stage.

4.3.5. V24 handler

Two transmission protocols are in use for the transfer of programs: the LSV2 protocol and the V24 protocol. This is the case, because not all the production machines have the possibilities to use the LSV2 protocol. For several machines it is too expensive to upgrade to the LSV2 protocol, for other machines there is no LSV2 protocol available. The communication with those machines is handled by the V24 handlers. Each of those machines has one V24 handler. The V24 handlers can only handle the transfer of programs. It is not possible to use the protocol for data acquisition purposes.

4.3.6. Digital data acquisition

The state of the production machines without the LSV2 protocol is monitored with the help of digital signals. The digital signals are collected by a combination of a system box and three digital data acquisition boxes all made by RWT. The system box is the controller of the communication. It has a connection with the production unit, with the terminal, with the production machine for the program communication, and with the MDE boxes. The other so-called MDE boxes are serial coupled to each other to transfer data to the system box. Each MDE box receives eight signals from the controls of the production machine.

If a signal alters, the MDE box notifies the system box, which in turn broadcasts the signal to the Fue Client process of the production machine. In this process the signal is handled the same way as the function keys. With the help of the programming language up to thirty-two signal actions can be defined, which are executed on the change of the signals. In our case the action is to send a message to the CNC server process defining the number of the transformed signal and the new state it transformed to (on/off). This message contains the name of the production machine, the name of the machine group, the time of the broadcast of the message, and the date of the broadcast. The server process distributes the messages to the relevant processes.

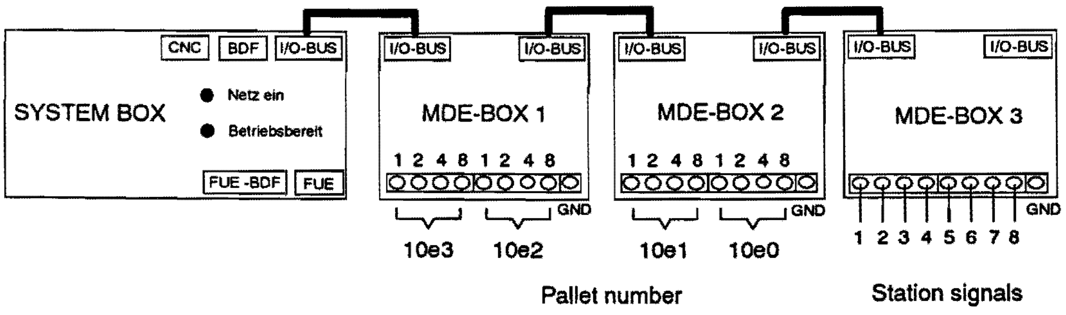


Figure 18: Configuration of the digital data acquisition boxes for the production machines

Two sets of signals are used, one for a transport unit and one for production machines. The signal set for the transport unit (figure 18) exists out of sixteen signals defining the four-digit pallet number in BCD code and eight signals to define the station. Each time a pallet enters a station, the controls of the transport unit trigger the signals defining the pallet numbers and one signal to specify the pallet station.

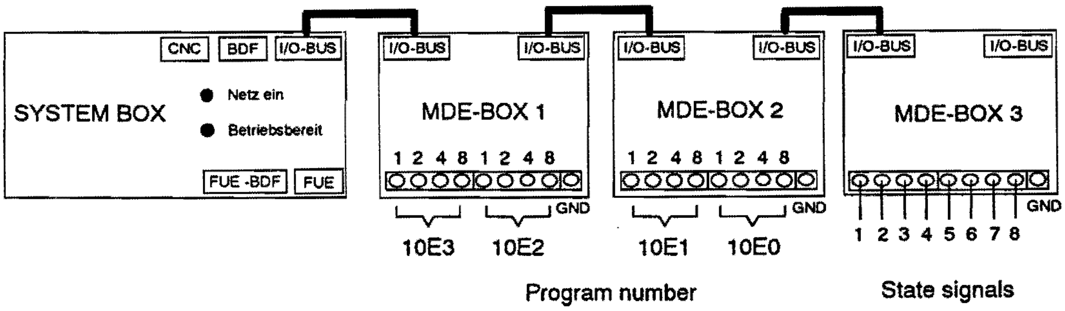


Figure 19: Configuration of the digital data acquisition boxes for the transport system

The signal set for the production machines (figure 19) exists out of sixteen signals defining the currently running four-digit program number in BCD code, and eight signals defining a change in state. These signals are:

1 End of program (impulse)
2 Program running (continuous)
3 No program running (continuous)
4 Production machine in automatic mode (continuous)
5 Production machine in maintenance mode (continuous)
6 Production machine control is running (continuous)
7 Fatal error detected (continuous)
8 Warning in effect (continuous)

5. Results

The monitor system has been tested with the help of a simulator containing the TRAUB TX8D controls. After the tests the system has been implemented for the Böhringer machine group and the Scharmann machine group. This system has been used for several weeks. The following sections describe the initial results.

5.1. Terminal handler

The terminals caused several problems. The largest problem is the fact, that the status of the function keys is identical for all production machines connected to the same terminal. If a function key is pressed, the terminal process executes the correct commands and generates the correct messages. However, if the display is switched to one of the other machines connected to the terminal, the status of the keys remains the same. This means for example, that when a job start is reported for one machine, the terminal subsequently shows the "job running" status on all production machines. Luckily the production computer receives the correct changes.

Another major problem is the sensitivity of the terminals for incorrect use. The terminal processes crashed regularly because of unintended errors while inputting data, such as pressing the wrong key. To restart the processes, the complete production unit has to be rebooted. During this restart, which lasts up to half an hour, the monitor system is unable to transfer any data to and from the production machines connected to the production unit. Therefore, all changes in the states of the resources and jobs are undetected and the available data on the states of the resources and jobs is invalidated.

Due to the large amount of processes running on the production units, the terminals react slowly. Delays of several minutes have occurred between the command for the listing of a program and the actual listing of the program. This time span is too large to be acceptable for the machine operator.

It turned out, that the terminal programming language is too limited in its capabilities. Firstly, there are only limited possibilities to update data in the database. Although the programming language has commands available to set the value of a field in the database, it contains no commands to increase or decrease the value of a field. Therefore, the terminal processes can display messages reporting the finish of a program, but is not able to increase the workpiece counter counting the number of finished workpieces. Secondly, the programming language has no possibilities of testing the input. Therefore, it is impossible to warn the machine operator after the input of incorrect data.

The error handling of the terminal processes is too limited. Although complex error messages detailing the errors are generated and logged into error files, the machine operator has to be satisfied with cryptic messages, such as "the handler finished the transfer incorrectly", which is generated with all sorts of errors during the transfer of a program. Therefore, the machine operator has no indication what went wrong.

Due to these problems, the terminals are only used for the transfer of programs between the production machine and the production unit. This severely limits the capabilities of the monitor system.

5.2. Communication processes

5.2.1. Digital data acquisition

The use of the digital data acquisition boxes resulted in several problems. Those problems can be split in three groups: one with general problems, a second with problems specific to the production machines, and a third with problems concerning the transport unit.

The general problems are all related to the handling of the signals by the FREENET software and hardware. To make it possible to use the same handling as the function keys, the FREENET software only acts on changes of the individual signals. It is not possible for any process to ask for the value of a specific signal. Therefore, all signals have to change after a start-up of or break of connection with a production unit, before the state of each signal is known. As it is possible for a signal to remain unchanged for several days or weeks, it can last a long time before the states of all signals are known.

The XENIX operating system of the production units switches between the competing processes in order to distribute the available processor time. Because of the number of processes running simultaneous, the execution of the processes slows down. The change of a signal triggers actions in a number of processes. After the change of a signal, the data acquisition box has to detect the change and report the change to the system box. In turn the system box, the related digital handler, the network handler, the network coordinator, and the CNC server have to detect and report the change of the signal. At the end of the chain, one of the Fue Client Processes receives the signal and sends a message to the production computer. This message contains a time and date of creation. As the creation of the message can be several seconds after the change of the signal, this date and time can differ from the time of change of the signal.

In a normal situation this time difference causes no real problems. However, if several related signals change at the same moment, it can cause problems. The system box tests the signals circular. Combined with the delay in the handling of the signals, a change of several signals at the same moment can result in differences in times of up to a minute. As it is possible for signals to change within a second, it is impossible to detect, wether the changes are related to the same event.

The second group of problems is related to the production machine controls of the Scharmann production machines. The signal "Automatic runs" changes each time a movement of an axis in the production machine starts or stops. Therefore this signal changes so often, that it is unusable.

Contrary to the definition of the signal, the signal "fatal error" is not immediately and not always followed by a stop of the production machine. This makes it possible, that this signal has disappeared before the resulting stop of the production machine is signaled. Therefore, it is impossible to detect, whether the stop of the production machine is caused by the previous error.

The set of signals used to detect the number of the program running on the production machine, also reports the sub program numbers. As those numbers are not related to a job, it is impossible to identify a job with this program number. Therefore, it is necessary to use the terminal to define the jobs running on a production machine.

The "end of program" signal is not always reported by the production machines. Therefore, it is impossible to detect the number of produced workpieces. Another problem is, that the "end of program" signal is not followed directly, if at all, with the "program stop" signal. Therefore, it is impossible to detect with the combination of these signals, whether the program has stopped after the previous workpiece is finished, or the program has restarted with a new workpiece and has been interrupted.

The signals gathered by the data acquisition boxes connected with the transport system can not detect, if a pallet is loaded to or from a production machine. Therefore, it is impossible to use the pallet number in order to connect program number, pallet number, and job number.

5.2.2. V24 handler

It turned out, that the program transfer is difficult with the V24 protocol. The machine operator has to start the transfer on the production machine and on the terminal. If the terminal and the production machine are located away from each other, this causes much inconvenience, especially because the control times of the V24 protocol result in the break of the connection, if the transfer is not started in time on both stations of the transfer. A solution could be the increase of the control times, or an increase of the number of terminals in the machine group. The increase in the control times results in long waiting times in the case of accidental start of transfers. As the waiting time can reach several minutes, this solution maybe unacceptable.

5.2.3. LSV2 processes

The LSV2 protocol between the production unit and the production machine controls is not able to gather all data necessary to the monitor system. Due to the concepts of the developers of the production machine controls, the availability of data may differ in different modes of operation. The LSV2 protocol of the Böhringer machine and the TRAUB simulator are developed for use in a completely automated unit. If those protocols are used in a unit, which is controlled by a machine operator, not all the data is available. An example is the TRAUB control, which reports the number of the running program only if the program has been started by the LSV2 protocol.

The LSV2 protocol of the Böhringer production machine does not report the start and end of the execution of a program. Instead, it reports the start and end of the manufacturing sequences of a pallet. Although the production machine has two pallets, during testing only one pallet has been used. All the necessary setup sequences have been executed inside the milling area of the machine. As a consequence, no messages have been send by the protocol, signaling the start or the end of the manufacturing sequences. If the LSV2 protocol is to be used successfully, the setup sequences have to be executed outside the milling area.

The program transfer with the Böhringer production machine still is not functioning correctly. The LSV2 protocol of this machine is identical to the Sinumerik LSV2 protocol. Therefore, it uses reaction and continuation telegrams in the transfer of data. During testing, the BCC code of the continuation telegrams generated by Böhringer production machine were incorrect. Therefore, it is impossible to download a program from the production machine. Transfers in the other direction stopped as a consequence of error messages in the reaction telegrams. The reason for the error messages could not be found and therefore could not be solved. To solve these problems, further testing of the communication is needed in cooperation with the manufactures of the Böhringer production machine.

The communication between the LSV2 handler and the CNC server of the production unit contains some errors. If the commands are triggered by the terminal, the commands for the transfer of programs to and from the production unit do not list the program number. The LSV2 protocol needs this number to initiate the transfer. In the case of the upload of a program to the production machine, the absence of the number forms no problem, because the number can be filtered from the program file. In the other direction the absence of the number blocks the transfer, because it is not possible to find the program number in any other way. As a temporary solution, the program number is hard coded in the case of the download of the program.

If the transferred is triggered from the production machine, the handler has to reject the transfer. The handler has no commands to ask the CNC server for filenames into which programs have to be stored. It is not allowed to generate an arbitrary filename, because the FREENET software uses coded filenames and will not handle filenames generated by other processes.

The transfer of programs is easier, compared with the V24 procedure. The transfer can be executed completely at the terminal. Only the deletion of programs on the production machine has to be done at the production machine controls, because of the absence of a command in the FREENET software.

5.3. Monitor process

If the CNC server detects a break of the connection with the production computer, it stores the messages for the production computer in a database. If the connection is reestablished, the production unit sends the stacked messages to the production computer. However the production unit does send these messages with considerable pauses. A comparison of the length of these pauses with the normal pauses between messages suggests, that the production unit sends the messages on a first-in-first-out basis, storing new messages at the bottom of the stack and sending an old message instead. This means, that for all production machines connected to this production unit, the monitored state lags in comparison to the actual state of the production machines. The problem is intensified, because the monitor system can not detect the break in connection and has no possibilities to force a production unit to send all stacked messages.

The monitor system is not able to detect the failure of a production unit. Only a restart of the unit is detected. Because the changes in the states of the resources connected to this unit remain undetected during the failure, all data available at the moment of the crash is rendered useless. Because the monitor system is dependent on other processes for the data and has no possibilities to request data, the monitor system has to wait for the changes in the states of the resources in order to reconstruct the states of these resources. When correctly functioning programs are running on the production machines the time between status messages can last several hours. Therefore, it can last several hours before the system is up-to-date. Especially for the digital signals, which can be unchanged for days or weeks, a solution has to be found to update the monitor system manually.

Because no connection is available to the database on the host computer, all data is stored in files on the production computer. In two directories, one for the machine status messages and the job status messages each, the messages are stored in separate files for the specific production machines. Due to the number of messages received by the monitor system, these files tend to grow fast. Therefore, these files have to be cleared regularly. The number of messages is already that high, that it is to be expected, that the capacity of the network and the database will be severely strained, when the complete production department is connected to the monitor system. This aspect requires major attention.

The master job list is transferred as a file between the host computer and the production computer. This transfer takes place after a planning run of the PPS-system. This procedure causes two problems. Firstly in the ten-day master job list several jobs may be listed with the same program number. Therefore it is impossible to link a program number to a job, unless the procedure is changed in a way that conflicting program numbers are non-existent. Secondly rush jobs and rework jobs are not included in the master job list. Currently it is not possible to add these jobs to the master job list. This means, that it happens, that a job is running on a production machine, which does not have an entry in the master job list. Therefore, the terminal has to be used to identify the job.

As a consequence of all the problems, the monitor system can generate only limited overviews of the states of the production machines. In the case of the Scharmann machine group, it is possible to detect the mode of operation, the operating status, and the existence of warnings or alarms. In the monitor files the changes in the mode of operation are listed with the date and time of the change, and the time elapsed since the last change. The warnings and alarms are logged with the start and end times. Changes in the program number are logged with the time and date of the change. The start, end and interruption of a program are logged with the program number, the time and date of the change, and, in the case of the end or interruption of a program, the time elapsed since the previous change in the program status. Because it is impossible to detect the replacement of a workpiece, the total manufacturing times can not be calculated, if a program is interrupted. Otherwise the time specified at the end of a program is equal to the manufacturing time of the program. In the case of the Böhringer machine group only warnings and alarms can be detected, due to the problems with the LSV2 protocol.

5.4. Tool handling

Although the tool handling has not been implemented, some problems have been detected concerning the tool handling. In the case of the production machines without LSV2 protocol, it is impossible to automatically generate cutting tool difference lists, because the contents of the tool storage magazines can not be checked. This data, together with the tool life data of the cutting tools, has to be entered manually by the machine operator. This introduces a great risk of errors.

In the case of the production machines with the LSV2 protocol, the differences in the format of the tool data and the identification methods of the tools may result in problems for the tool handling methods. This should be a major concern in the development of the new handling methods.

Currently there is no information available to determine the required tool life of a cutting tool needed within a program. This information will be important for the planning system, in order to execute a detailed scheduling of the cutting tools. Therefore, a solution has to be found to include this data in a program.

Another problem is caused by the absence of a detail planning of the jobs. Therefore it is difficult to generate difference lists for the resources. If a job combines several parts, it maybe possible to use an auxiliary tool several times within the job. For example, if more than one identical workpiece is manufactured on the same production machine, it may be possible to limit the number of fixtures to two, instead of the number of workpieces, if a fixture is reused several times. At the moment it is impossible to check for those possibilities.

6. Conclusions

The available FREENET system is unusable in the current hardware configuration. The production units are too slow to execute the necessary processes at an acceptable speed. When jobs are running on several production machines connected to the same production unit, the time between request and result can be several minutes. It is necessary to make a choice between fewer production machines per production unit or faster computers.

The existing programming language has to be changed. In order to make a better use of the possibilities of the system, it is necessary to have more influence on the data available in the INFORMIX database. Especially improved tools are needed for changing data when receiving signals and for checking the input on errors.

The bugs in the FREENET software severely limit the capabilities of the monitor system. The major problems are the display of the state of the different production machines in a machine group, the broadcast and safeguard of data in case of a lost connection, the commands for and the handling of external device drivers. Due to the bugs, it is not possible to use the terminals to gather data manually with the help of function keys. These bugs also limit the possibilities of program transfer between the production machine and the production unit.

In the communication between the production machine controls and the production unit several problems exist. In case of the V24-communication the way of transmitting the data makes it necessary to have the terminal near to the production machine. The data received from the digital MDE boxes can be used only for the detection of the state of a production machine.

Due to the problems with the program numbers, it is not possible to couple the state of a production machine to specific jobs. The same holds for the LSV2 communication. This is due to the limitations of the Sinumerik and TRAUB TX8D controls. This means, that the monitoring system is heavily dependent on manual input.

A major problem will be the cutting tool handling. For the production machines monitored with the digital MDE boxes all data has to be entered manually. The differences in the LSV2 protocols may result in identification problems of the cutting tools in the tool storage magazines.

If the performance of the FREENET software and hardware does not increase with the next upgrade of the system, it is recommended to change to an other software package. The current performance is unacceptable and limits the capabilities of a monitor system to almost the same level as the old terminal system.

Reference list

C. Schlotz, M. Röck, R. Baisch
Projekt PFIF: Bedieneroberfläche FREENET Terminal, internal report TRAUB AG, 1990

C. Schlotz, M. Röck, R. Baisch
Pflichtenheft Projekt PFIF, internal report TRAUB AG, 1989

IBM
Programming manuals AIX 6000, IBM, 1989

Microsoft
XENIX development package, MICROSOFT, 1989

RWT GMBH
CNC server manual, internal report RWT GMBH, 1990

RWT GMBH
FREENET manual, internal report RWT GMBH, 1990
Siemens
Manual Sinumerik 850, Siemens, 1989

Siemens
Manual Sinumerik 880, Siemens, 1989

TRAUB
Betriebsanweisung TX8D Steuerung, TRAUB AG, 1987