CHALMERS

EFFICIENCY MANAGER

DEVELOPMENT OF AN OEE MEASUREMENT SYSTEM

Jon E. Andersson

DEPARTMENT OF SYSTEMS, CONTROL AND MECHATRONICS
Division of Automation Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2010
Report No. EX047/2010

Report No. EX047/2010

EFFICIENCY MANAGER

DEVELOPMENT OF AN OEE MEASUREMENT SYSTEM

JON E. ANDERSSON

Department of Systems, Control and Mechatronics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2010

Efficiency Manager
Development of an OEE Measurement System
JON E. ANDERSSON

Technical Report No. EX047/2010
Department of Systems, Control and Mechatronics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Göteborg, Sweden, 2010

Abstract

To support production sites to calculate the efficiency of their operations, software systems can be used instead of manual studies. A software study compared to a manual study can be continuous and able to follow trends in efficiency. A common efficiency measuring method is OEE (Overall Equipment Efficiency). This measuring method uses information about available production time, planned stops, unplanned stops, production pace, produced amount and wasted amount. This master thesis project specifies and partly implements a system able to collect and present OEE-data. The system aims to, by small effort from the operators and small influence on the production, collect and present OEE data.

Very important byproducts from the collected data presented in this report, are information about stop reasons and waste reasons. By using statistical tools, the system is able to provide reports of, for example, the most frequent stop reasons or trend diagrams of wasted materials.

The report describes the required data to gather and how to add important information with a small effort from the operators. The result is an expandable, flexible system that can be implemented in multiple types of production environments. The requirements on the monitored production part are to provide the current production pace and the result of an automatic or manual quality check.

Keywords: OEE, Software Development

Acknowledgements

The project had a great support from many employers at AcobiaFLUX. I would specially like to thank my supervisors Anders Jönsson and Martin Esping for invaluable support and feedback throughout the project. I also want show my gratitude to my examiner and supervisor Prof. Martin Fabian for his interest, support and feedback of the project, especially in the report editing phase.

Table of Contents

Abbreviations

- OEE – Overall Equipment Efficiency
- TPM – Total Productive Maintenance, set of methods to increase productivity.
- GUI – Graphical User Interface
- ER-diagram – Entity-Relational diagram
- SQL - Structured Query Language

1 INTRODUCTION

For production sites where lots of money and effort are put into process improvements, a focus on processes that will benefit the most from the improvements is important. To trace the best spots for improvements it is essential to get reliable information about the processes. Information about the processes is commonly retrieved through studies and statistical analyses, but an information gathering activity may be time consuming and costly. If the operators gather information by themselves continuously, loads of information can be retrieved. However, a poorly implemented reporting system may have major impact on production performance. Thus, systems that collect information continuously but still have a low impact on production are very valuable.

To provide a solution to retrieve the information with small impact on production, this report recommends further development and usage of a system called *Efficiency Manager*. *Efficiency Manager* retrieves data and provides information and numbers about the performance in the process. The system measurements are based on OEE (*Overall Equipment Efficiency)*. OEE is a widely used key value for efficiency measurements. Nakajima, S. (1988) proposed OEE as a measuring method in his work about *Total Productive Maintenance* (TPM). The three components of OEE, *availability*, *performance* and *quality*, incorporate measurable and important factors of the process.

The basic idea is to retrieve information about the production pace from the existing production control system. *Efficiency Manager* is then able to decide whenever the process has stopped, is operating and in what pace it is operating. The operators categorise breakdowns on a touch screen from predefined categories. That information together with data of wasted materials, also reported manually on the touch screen, gives important information to help production engineers to find inefficient processes to focus on. The spin-off from OEE-measuring is statistics about critical factors of the process. Factors that are measured in *Efficiency Manager* are breakdowns and wasted material both together with a reason for the occasion, furthermore is the produced amount and the production rate stored. For analysis of the stored data, the information may be mapped on different processes, different products or time. The simple measures done by the system gives lots of important information to the production engineers in the production facility. OEE reveals the capacity of the production system.

1.1 PURPOSE

The root cause for the initialization of Efficiency Manager was three existing similar prospects. By the start of this project none of the three prospects was very close to order. It was instead decided to make a general system that can be used as a base for the prospects. The system is then possible to adapt to the costumers own requirements.

Efficiency Manager is intended to help engineers to continuously get accurate information about problems in the operations. The information gathered can be used both to find problems and to follow up improvements. The report specifies how *Efficiency Manager* should be implemented, and implements major part of system. This provides a good base for further developments of the system.

The basic idea of this system is to retrieve information for the OEE measurement by minor effort from the operators and not influencing the production. By adding some information to the minimum OEE information set, lots of additional important information is gained about the system

1.2 Objectives

The project shall make an analysis and state an example with needed functions and requirement for the system. This is presented in a requirement specification; see Appendix A (in Swedish).

The system should be easy and fast to use for the operators and administrators. The system should not hinder the users in their daily work.

The system should be able to provide statistics about the OEE-value in the processes. The individual required components for the OEE-calculation should be possible to examine.

Breakdowns and wastes should be able to categorize. The categories are different reasons for the breakdown or waste and should be used to trace problems in the processes.

Major parts of the system specification should be implemented. The implementation should include the operator's interaction through a graphical user interface, intended to use on a touch screen.

1.3 Method

The project has two development phases with a big overlap. The specification is developed in the first phase. The activities for the stage include discussions with experienced developers, usage of experience from previous similar projects and some test of graphical proposals. Multiple parts of the specification and the implementation are clearly inspired from existing projects. One important inspiration is the product *AX CARAT*. *AX CARAT* is a tool used to track stops in production. It uses one signal to decide if the production is on or off. If the production is stopped the operators have to tell the system why it stopped. The idea of *AX CARAT*'s stop registration is modified and used in *Efficiency Manager*. The second big inspirations are the three prospects that were the initiators for this project. Those are internal documents and will not be referenced. (AcobiaFLUX, 2010)

The second phase of the project concerns implementation of vital parts of the system. The server is implemented in a very early stage and includes a SQL-server and a service described later. The early implementation of the SQL-server enables decent and reliable feedback for further development and testing. The SQL-database defines what type of information to be stored. Parallel work with the SQL-database and the specification make it possible to visualize and test the specification. It results in a reliable SQL-database and specification that does not requires major changes during the remaining implementation phase. Late major changes in the SQL-database can result in larger unwanted changes in the rest of system.

The development of the *Reporting Client* does also help to visualize and test the ideas for the specification. After implementation of vital parts it is easier to visualize the system. The discussion with the supervisors for the project is more valuable when parts of the *Reporting Client* (Section 3.5) and the rest of the system are implemented. Experienced developers give better feedback when the system and the ideas are visualized.

2 Technical Framework

2.1 Total Productive Maintenance – TPM

Japan took a new turn in the industry sector after the Second World War. To be competitive they had to improve the quality of their products. With help from the United States, Japan adopted new methods called *Preventive Maintenance* (PM). The Japanese modified it and developed their methods, called TPM. As a part of the internationally known *Lean production*, TPM was a big factor of the Japanese industrial success. The quality and productivity in the Japanese industry became world famous in the later part of the 20th century. (Nakajima, 1994)

2.1.1 The six big losses

The purpose of TPM is to eliminate the *six big losses.* (Nakajima, 1994)

Breakdowns

The first of the six big losses is breakdowns. Losses included in breakdown are time losses and waste from defective products caused by longer sporadic breakdowns. Personnel working in the industry spend a lot of time investigating the causes from big breakdowns, but eliminating all the breakdowns is difficult. To reduce breakdowns proactive maintenance is an important tool. The maintenance makes sure that the equipment is operating as intended. Some maintenance will cause the production to stop. People will claim the stop is as bad as breakdown, so why bother about high level of maintenance? The maintenance activity is planned and can be executed in planned production stops. In a planned maintenance activity all materials are in place, the production can be stopped with safe and controlled approaches. The main difference between a planned stop and an unplanned breakdown is the production time saved while planning the repairs. (Nakajima, 1994)

Setup and adjustment

Setup losses derive from stoppage time and defects occurring when the machine is adjusted for a new product. Setup can often be significantly reduced. Thus, big setup losses are a big potential for improvements. To cut down setup time, the engineer separates the setup activities that can be executed during run time and strive to minimize the time for the rest of the activities executed during production stop time. (Nakajima, 1994)

Idling and minor stoppage

Minor stoppages are short stoppages which can be resolved quickly and easily. They are separated from breakdowns because of the difference in how operators handle them. Minor stoppages has a major effect on the performance but is often hard to track. Problems that are hidden among other problems can be neglected as small problems. If the problems are hard to visualise they are hard to get rid of. When minor stoppages are visualised and quantified they are easier to correct. Tools used to track and store information about the stoppages in production can be useful to visualise the problems. (Nakajima, 1994)

Reduced speed

When the design speed (the maximum speed that the machine is designed for) for the equipment is higher than the operating speed, there is a constant loss in efficiency. The reason for low production speed originates from other problems in the production. Reduced speed has its cause in bad raw material, worn out machine parts or previous breakdowns from high speed. Consequently, the operations have to be secured to be able to raise production speed. Design speed will not be achieved before other problems in the operation have been mended. (Nakajima, 1994)

Quality defects

Defects requiring rework or recycling is one of the worst losses. The wasted products are using time and costs through the process but do not yield any value. The work done in previous operations have to be redone. More quality inspections reduce the risk of bad products using further resources, but inspections are costly. A better approach is to find and eliminate the source of the quality problems. Quality problems commonly have their origin in an unstable process with many stops. By securing and eliminating problems in the operation, the quality will rise together with performance. (Nakajima, 1994)

Start-up losses

Losses occurring from start-up as scrapped products, test products and products suffering from unstable processes in the start-up phase are bigger than expected in many cases. The losses are many times, according to Nakajima (1994), neglected as unavoidable. The losses have big potential to be reduced and make the processes obtain good efficiency.

2.1.2 Overall Equipment Efficiency – OEE

To find a way to summarise the losses, Nakajima proposed to use OEE. OEE is a multiplication of the three components *availability*, *performance* and *quality*. The three components are mutually exclusive and offer three important key values for the measured process. Some argue that OEE does not give the big picture of the factory. Scott and Pisa (1998) claimed that one should look beyond the performance at equipment level and see the big picture at factory level. It is important to know the limits in the measurements and use it for the intended purpose. Other methods for performance measures on the factory level have been developed. For example, Muthiah and Huang (2007) developed a method. The method is an adaptation of OEE to the factory level but does not take into account if one part of the production is inefficient, as long as it does not constrains the factory.

As Jonsson and Lesshammar (1999) discuss, it is hard to use OEE to compare different firms and shops, partly because it does not cover the whole picture of the system. OEE measures efficiency and does not consider economics, environment or social health. Consequently, it is possible for a very efficient process to have a high OEE-value but not be as profitable as another process with lower OEE-value. Partly it is hard to compare different firms and shops because of variations in data collection methods and calculations. Manual data collection does not give as good accuracy as automated gathering, e.g. Jonson and Lesshammar (1999) pointed out the problem with data collection performed by personnel that affecting the parameters. They tend to make the numbers better than they really are. This shows the need of standardized measuring methods throughout the company. The information should not be affected by the personal that influence the measuring parameters.

Comparing is not the preferred application of OEE. The application for OEE is to use it as a tool in improvement activities of processes. The components in the OEE calculations give feedback after the improvement activity by showing the outcome on the OEE components. A constantly updated OEE value is also a direct feedback to the operator how good today's work is. The operators get a value that explains how efficient their work method is. The methods may then be improved without contribution from a production engineer. However, it is dangerous to use the OEE as a performance value of the operators, since they may feel superintended. Consequently, the operators will then most likely not support the measures. Instead OEE should only be used to investigate general methods in each measured area over a longer period and with multiple operators, rather than on each individual.

Because of the problem to compare different OEE values, it is difficult to provide common results for OEE. However, OEE values around 50 % and lower is not unusual. World class OEE is asserted by Nakajima (1988) to be over 84 % with the component values of Availability greater than 90 %, Performance greater than 95 % and Quality grater then 99 %.

An example of good, not world class, OEE-values is displayed in Figure 1.

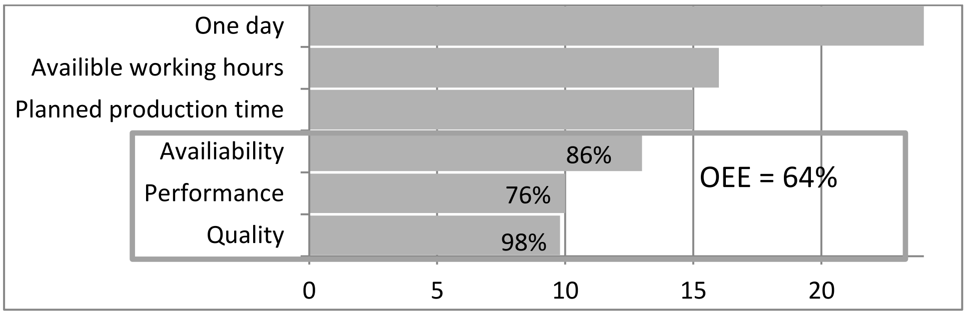


Figure 1 Example of a typical OEE-calculation. 86 % of the planned production time is used, 76 % of the avalible time is used and 98 % of the produced products are used.

Availability is affected by the first two of the six big losses, breakdowns and setup and adjustment, see Equation 1.

Equation 1 Calculation of availability; one component of the OEE equation.

$$Availability = \frac{Planned\ Production\ Time - Stop\ Time}{Planned\ Production\ Time}$$

Performance is reduced by small stops and lower production speeds, the third and fourth loss of the six big losses, see Equation 2. The ideal cycle time is important and sensitive value. If the cycle time is set too long, the performance sometimes go beyond hundred percentages when the real speeds exceeds the ideal time. High OEE-values lowers the incentives for improvement. If the cycle time is set too short the performance seems to be a bigger problem than it is. To calculate a good and fair value there are, according to Nakajima, three methods to estimate cycle times by.

- Use the designed speed for the machine. When the machine has a highest speed value in the specification it may be used as the ideal cycle time.
- Use cycle times that are based on current optimal conditions. Ideal cycle time can vary according to produced product or used equipment.
- Use the best cycle time achieved historically.

Equation 2 Calculation of performance; one component of the OEE equation.

$$Performance = \frac{Ideal\ Cycle\ Time * \ Produced\ Amount}{(Planned\ Production\ Time - Stop\ Time)}$$

Quality is reduced by the wasted product from start-up and production problem occasions, see Equation 3.

Equation 3 Calculation of quality; one component of the OEE equation.

$$Quality = \frac{Produced\ Amount - Wasted\ Amount}{Produced\ Amount}$$

3 SYSTEM SPECIFICATION

The system specification explains the functions and interactions in the system of *Efficiency Manager*. Appendix A provides a full specification of the system written in Swedish. The following Section contains a shortened description of the most important parts in the system specification. The total system specification is not implemented, but main components including vital server parts and a *Reporting Client* is implemented. Chapter 4 describes in detail the design of the finally implemented parts of the specification.

3.1 SYSTEM OVERVIEW

Efficiency Manager is built to be easily expandable. After installation of the hardware the system administration process should be easy and intuitive. The administration of the system should be possible to be performed by the company themselves. This minimizes the time from ideas for revisions to implementation of the ideas. It also helps understanding the system and how different parameters affect the resulting statistics.

The architecture is a *server-client system*, where all information is stored in a central server and the client calls the server for read or write operations. The clients do not communicate directly with each other; instead every client polls the server for new information. The concept is displayed in Figure 2.

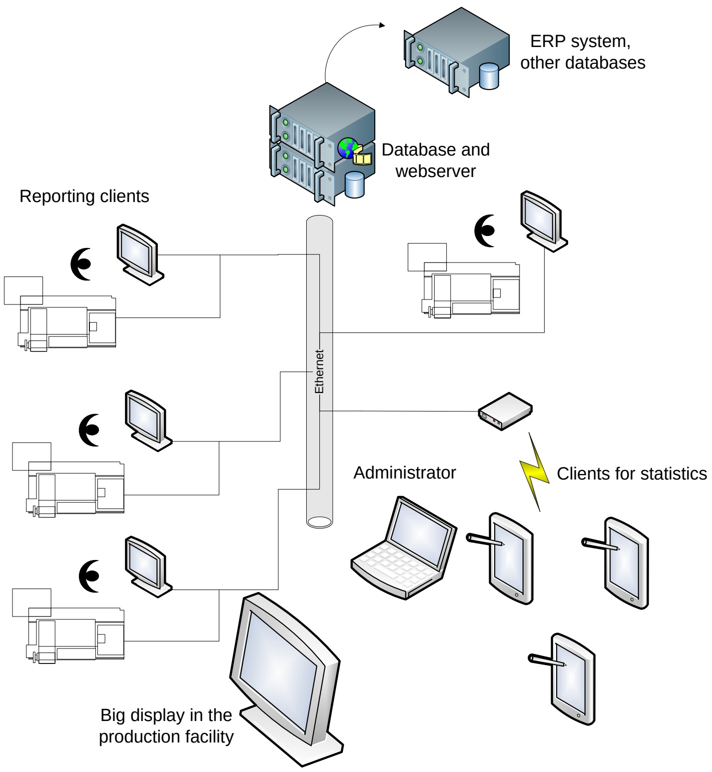


Figure 2 System overview.

3.2 BASIC IDEA

The calculation of the OEE value requires the parameters *available production time*, *planned stops*, *unplanned breakdowns*, *production speed* (small stops lowers speed), *theoretical speed*, *produced amount* and *wasted amount*. To support the idea of a flexible system, three base structures for storage are defined; *production order*, *breakdown* and *waste*. The three storages are separated from each other and can be registered independently, but a stoppage or a waste registered during a time when no order is defined will not be included in the OEE calculation. Wastes registered after an order is finished have to get a time at the end of the order, in order to have a loose connection to the specific order. The loose connection between the three structures makes it easier to enable different approaches to log data, e.g. use automatic waste registration and let another system store wastes.

3.3 SERVER

The heart of the system consists of an SQL-server (*Structured Query Language, Microsoft SQL-server 2008*). In the server both system information and statistical log information are stored. Access to the SQL-database is performed internally in the server by a *Windows Service*. The windows service acts as an interface to all clients in the system through the *HTTP* protocol. Within the service, it is possible to define what information to expose through the intranet. It is also possible to define how information is added. The clients call standard methods providing information in parameters, there is no need for SQL-queries from the client side. The windows service executes the required SQL-queries and returns the requested value. A change in the database structure results in changes in the windows service, but not necessarily in all clients. The central service design results in an easy expandable architecture.

The server does not store any information about the clients or any connections. Hence, the stateless server is not affected by a fail in connections or new client calls. A stateless server never cares about what the client asked earlier. This makes the architecture strong against system failures. The only loss of information could be locally stored information in clients about current order or breakdowns in case of a client failure. It is also easy to change physical clients, thus any client can monitor any production part. Security aspects can be controlled by defined user groups for clients with different access rights.

3.3.1 SQL-SERVER

The SQL Structure is described as an *Entity-Relation diagram* (ER-diagram) in Figure 3.

The server has three tables for all its statistical data. The first data table consists of order data. Each order is represented by one row in the table. The standard information is *planned amount*, *produced amount*, *production times*. If the customer wants to extend the information stored about the orders, the table is extended with the required fields. The second logged table consists of information of the breakdowns. For each row in the table, information about the *reason*, *time*, and which *production part* is stored. In the table it is possible to add fields that describe the situation at the occurrence of the breakdown. The third table stores wastes. The wastes are stored together with *time*, the *reason* and *production part* identification. As for breakdowns it is possible to store actual environment variables at the time the waste was logged.

The server has one table, *production part*, which consists of base information about the system. The table and links to other tables describe the production parts properties. From these tables the clients are able to get the system tree, associated products to each production part and associated breakdown and waste reasons. The tables and their relationships are created through the administration client.

The server also stores templates for the system parts. The templates contain all the static data that the production parts contain, including associated breakdowns, wastes and products. The templates are used to provide a way to standardize many similar production parts. Hence, it is possible to use the template to add new reasons to all associated production parts. It is possible to change all associated production parts by changing the connected template. The templates could be used to quickly add or remove breakdown reasons, waste reasons or products for a group of production parts.

Manual changes are intended to be logged. A change log makes it possible to store who made the change, when the change was made and the value before and after. A change log is not made automatically in the SQL-server. The service provides methods for creation of change logs. These methods have to be called by the client application when the change to the SQL-server was manual, e.g. a change log method is called by the *Reporting Client* every time a breakdown reason is modified. The change log can be used by administrators to track modifications done to the system. It is possible to fully rollback a change using the information in the change log.

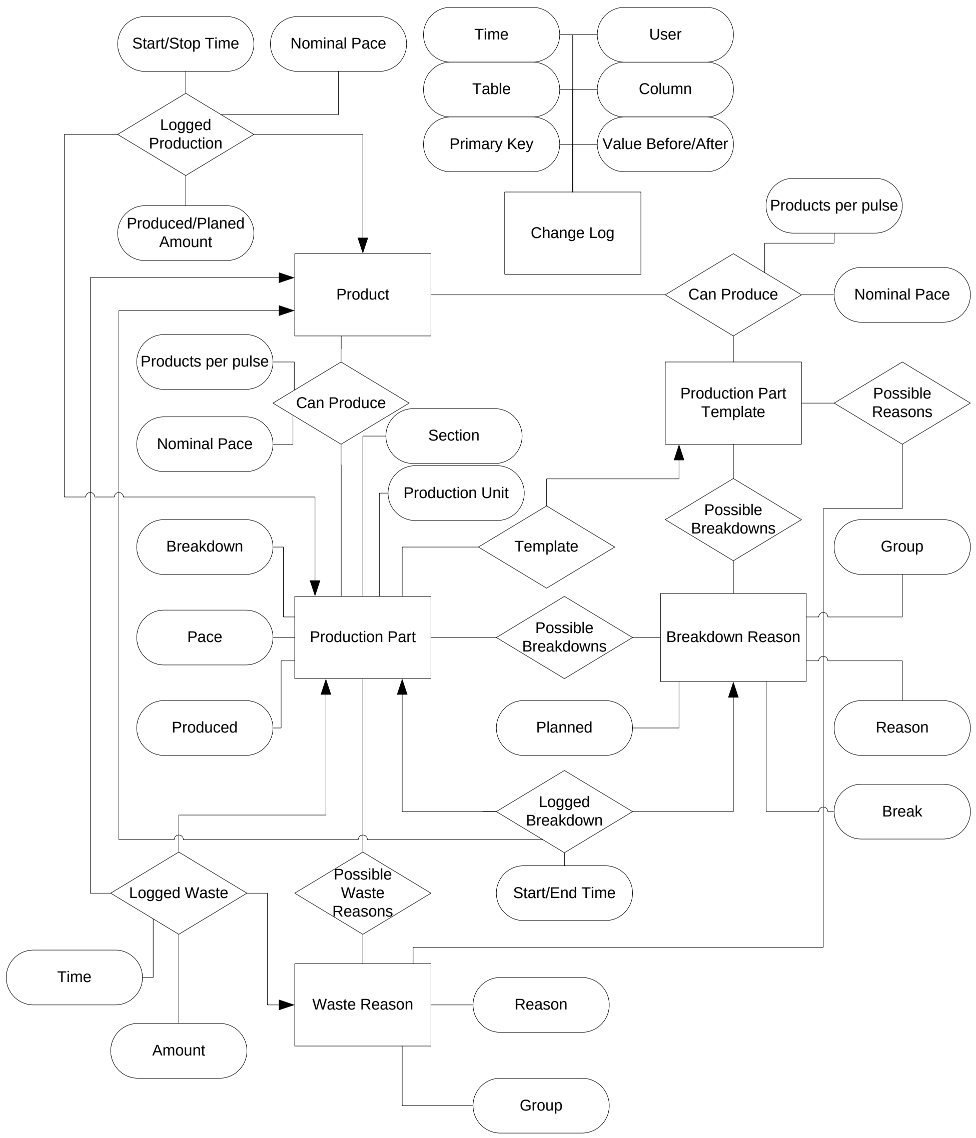


Figure 3 ER-Diagram of the SQL-Server.

Expansions

As explained earlier, it is possible to add logged variables for production, breakdowns and wastes. The expansions make it possible to track environment situations for the different activities. Expansions could increase the speed of finding root causes for the problems in the processes, e.g. if very high temperature always results in a breakdown. In the tables only one value for each extended measuring value can be stored. To handle multiple measuring points for a single order, a new table is created with a reference to the current order.

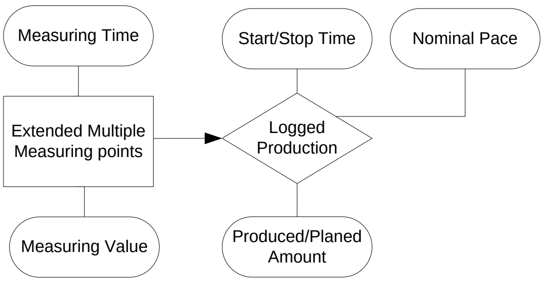


Figure 4 log extension with multiple measure points in time.

3.3.2 WEB SERVER

The server also contains a web server. The web server hosts the administration webpage and the *Report Generator's* webpage. In the webpages it is possible to restrict different sections to only be accessible for some users or group. The different pages are described in Section 3.4 and Section 3.5.

3.4 ADMINISTRATION CLIENT

To administrate the system, the administrator uses a regular web browser. From the web browser all *production parts*, *products*, *breakdowns*, *wastes*, *templates* and their relations are defined. The pictures in this chapter are examples of how it can be implemented. Figure 5 shows how the webpage is organized. The first part is an administration part and the second a report part. By clicks in the tree the user is able to use the features in the pages.

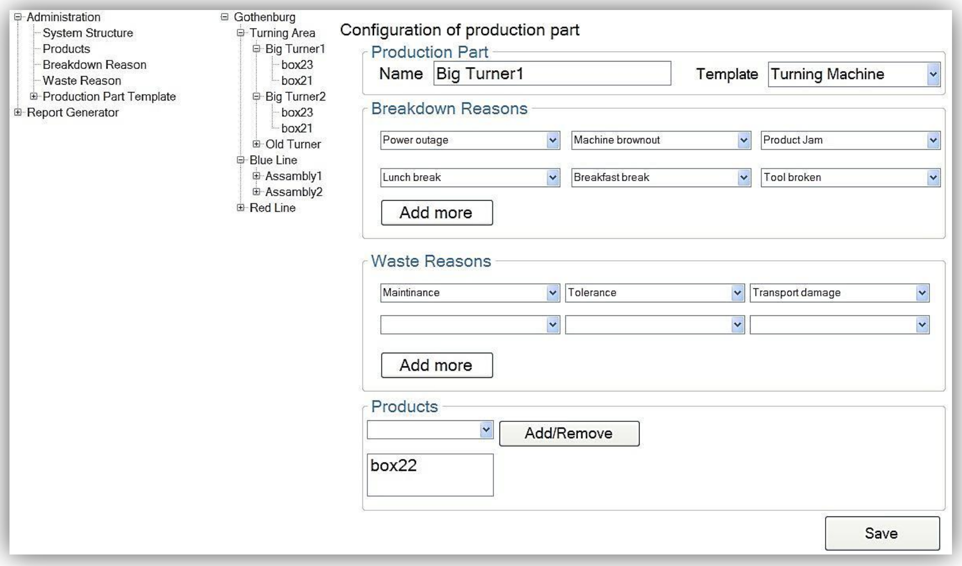


Figure 5 Configuration of a production part that exists in the section "Turning Area" and unit "Gothenburg".

3.4.1 System Structure

The user composes the system; the relations between the different components in the system are visualized in a system tree. The associations in the server are visualized as a parent child relation in the tree. In Figure 5 the system tree is visualized. A click on any entity brings up a page for configuration of the specific entity. Products are connected to production parts, e.g. "*box23*" in "*Big Turner*". Production parts are connected to sections, e.g. "*Big Turner1*" in "*Turning Area*". Sections are connected to units, e.g. "*Turning Area*" in "*Gothenburg*". In **Fel! Hittar inte referenskälla.** the production part "*Big Turner1*" is configured with breakdown reasons, waste reasons and products.

3.4.2 Template Definition

The templates are configured in an equivalent approach as a special production part. Templates are used to decrease the repetitive work for the administrators. When production parts are associated to a template, they are changed when the template is changed. A change at a template activates a question to the user if the associated production parts should be changed as well. This simplifies addition of e.g. breakdown reasons for a group of production parts.

3.4.3 Product Definition

Every product on every production part can be individually configured; however, the standard values from the template are loaded when the production part is associated with a template. When the product is added to a production part, the administrator can configure the product and set nominal production rate and products per pulse. See Section 2.1.2 on how to set the ideal cycle time, discussed here as nominal production rate (note that rate is the inverted cycle time). Products per pulse are the amount of products processed every time the *Process Interface* adds a product to the database. Figure 6 shows a product being configured in association with the production part "*Big Turner*".

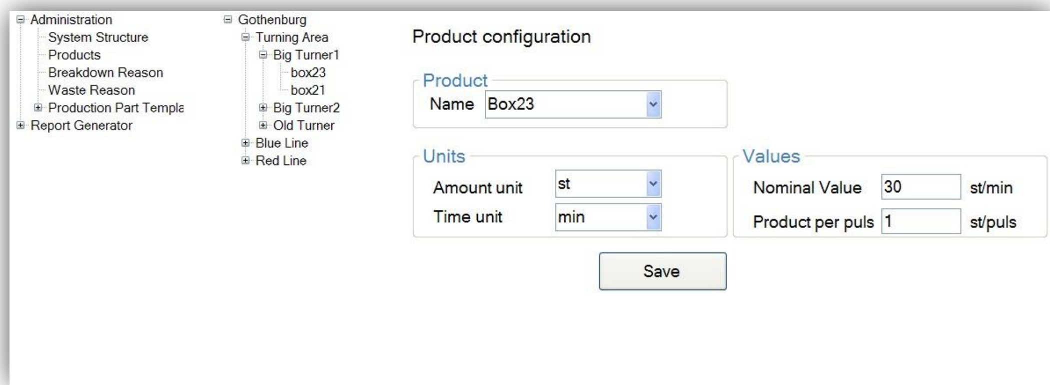


Figure 6 Configuration of a product in "Big Turner1".

3.4.4 Breakdown and Waste Definition

The definition of breakdowns and wastes is performed in separated tabs before they are added to production parts. They are defined by a name and if needed a group. If a group is defined, the *Reporting Client* asks the user for the group and then a reason. If no group is defined the *Reporting Client* shows the reason directly without any grouping.

Two special breakdown cases are available for breakdowns, see Figure 7. The user can specify if the breakdown is planned and if it is a break. A planned stop will be managed as a planned stop in the OEE calculations, see Section 2.1.2. Hence, planned stop time does not affect the OEE-value negatively. The breaks act differently to normal breakdowns.

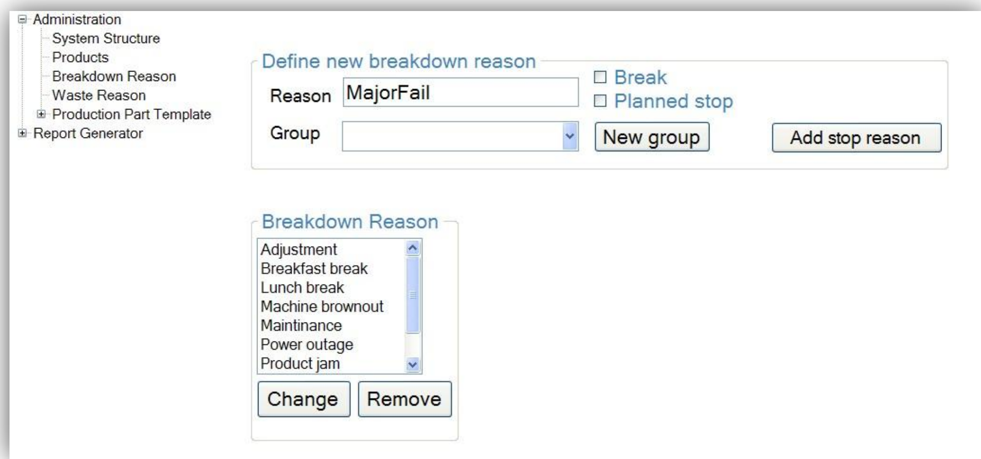


Figure 7 Configuration of a breakdown reason.

Breaks

Breaks are treated specially, i.e. breaks are often stipulated how long they are and when they occur. Hence, the break is not always as long as it should be. It is important for the statistics to get the real length of the break.

Another important issue is common activities such as adjustment or set-ups in conjunction with a break. It is important to separate breaks and such activities. To catch the different behavior of the breaks they are handled differently in the *Reporting Client* compared to other production stops. The default way to treat a break is to declare the break ended when the operator returns from the break. If the *Reporting Client* still reads a stop in the production, a new breakdown is generated. It is important that the break is defined before other activities starts, otherwise the extra activities are declared as break time.

Another way to handle breaks is to define a definite time for the break. When the time for the break is ended, a new stop is created. The operator is forced to choose a new reason for the extra stoppage time, e.g. adjustment or extra break. The disadvantage is that the system does not catch the repairs that occur when an operator returns earlier from a break.

Breaks are discussed further in Section 3.8.2.

3.5 Report Generator

It is possible to generate statistical reports about the production from a web browser. The *Report Generator* can configure a report for many specific needs. To expand the *Report Generator* when the system is extended to measure more parameters is easy. From the start, reports can be generated consisting of statistics for:

- Overall Equipment Efficiency
- Wastes - can be filtered on individual reasons
- Breakdown minutes for each part - can be filtered on individual reasons
- Breakdown occasions for each part - can be filtered on individual reasons

- Produced amount
- Breakdown reasons – amount of occasions for each breakdown

All information except breakdown reasons can be retrieved as a comparison for different production units, sections or production parts. Two different types of reports can be generated, either trend reports as a function over time or accumulated bar charts. All information may be retrieved as raw data or diagrams. The reports can be exported as *pdf documents*, *excel documents* or *word documents.* The reports are simple but show useful information.

3.6 Process Interface

The *Process Interface* will be implemented differently for different systems. It will be the layer to convert the signals from the control system and puts the correct data into the database. The signals can be provided in many forms. The minimum information from the signals must contain information about the current production rate, e.g. pulsed signals for every batch produced.

When the interface is under development it is of great concern that it shall be easy to install and not affect the process in runtime. Ideally the installation of the interface does not stop or affect the process in any matter. In a second best case the installation process is done during a planned stop. If the costumer does not want to extend *Efficiency Manager* to use it as a process controller, the system only reads information from the control system. A system that only reads existing signals does not affect other components. If the desired signals are available it shall be easy to install.

3.6.1 Specification

It is not this projects intention to specify exactly how the *Process Interface* executes or how the hardware is operating. The specification states a boundary that explains how the *Process Interface* will work. The core components of the *Process Interface* will be common for every installation. The difference will mainly consist of the hardware used and some software to convert the signals to the form wanted.

Once installed, the interface shall not require any attention from the operators. After installation it is running and monitoring the connected process. The *Process Interface* connects to the server using the *Windows Service*, Section 3.3. Furthermore, it is responsible to update the database through the service about the current state of the process. The interface does not directly influence the data that the statistics is based on. It only affects temporary variables for each production part. The information is later processed by the *Reporting Client* for statistical logging.

The base functions called by the *Process Interface* in the service are:

- *AddProducedAmount* (Production part identifier, Amount produced) – Adds the amount of produced products since last call. The value will later be multiplied by a value, products per plus (Section 3.4.3), by the *Reporting Client* depending on the currently produced product.
- *SetCurrentPace* (Production part identifier, Current pace) – Sets the current pace in the production. The value will be multiplied by a value, products per plus (Section 3.4.3), by the *Reporting Client* depending on the current produced product.

- *SetBreakdown* (Production part identifier, Breakdown time) – Sets the time the breakdown started. Zero if production is running.

When the *Process Interface* gets signals from the process showing that it is running, the breakdown value is set to zero. For each signal from the process that one piece or a batch has been produced, the interface calls the *'AddProducedAmount'* function and the *'SetCurrentPace'*. *'SetCurrentPace'* defines the current production speed and *'AddProducedAmount'* adds produced amount to the database. When the *Process Interface* perceives a stop in the process, a call to *'SetBreakdown'* providing the last time the process produced a product is called. There is a delay in the *Process Interface* before it calls *'SetBreakdown'*. The delay defines if the stop is a breakdown or a short stoppage. A short stoppage is defined in Section 2.1.1, as a short and easily resolved stoppage. Defining the breakdown delay must be done for each process. The delay can be set by experience of how long it usually takes for a small stoppage to be restored. Whatever value chosen, the final OEE value will be the same. The difference between a long or short delay is a drift of waste among the OEE components performance and availability. Thus, short stoppages are included in performance and breakdown is included in availability.

3.7 REPORTING CLIENT

At every monitored station the operators will use a touch panel client with a *graphical user interface* (GUI). Through the clients the operators performs all tasks required. The GUI has to be intuitive and easy to use. All operations will be done with few clicks in order to keep the operators to focus on the process.

The *Reporting Client* has a central role in the system. It determines when and what to store in the server. The *Reporting Client* repeatedly reads information in temporal storages within the database and uses the information from the touch panel to decide what to do. When the *Reporting Client* has got all the information needed for any log event, i.e. *breakdown*, *waste* or *produced order*, a row is stored in the SQL-database. The log activities are described in Section 3.3.1.

The operations that are performed in the *Reporting Client* are:

- Get OEE-statistics for the latest hours of production.
- Declare stop reasons.
- Register waste.
- Define currently produced product.
- Edit stop reasons.
- Edit and delete waste.
- Retrieve statistics about predefined key values, e.g. breakdowns.

In the customers production facilities there are probably some other systems partly or fully performing the operations stated here. *Efficiency Manager* should try to exploit information from those systems. It will probably require some work to get the information in a form that is usable for *Efficiency Manager*, but unnecessary work for the operators are saved.

One example of a possible reused function in another system is for the waste registration. An existing system for automatic quality checks can be used. Whenever the system gets an indication of bad quality, the existing system logs the information somewhere. Either the existing system is configured to also log the waste into *Efficiency Managers* SQL-database, or a program monitors the quality systems activities and stores the quality information into *Efficiency Managers* SQL-database.

The *Reporting Client* is implemented and is described more deeply in Section 4.2.

3.8 MODIFICATIONS

In order to fit *Efficiency Manager* into the existing production system, modifications will have to be made in *Efficiency Manager*. Most of the modifications concern the use of existing production system to hinder dual work for users. *Efficiency Manager* is built as far as possible to minimize the effect of the modification to the features and components that are directly affected.

3.8.1 MAKE USE OF AN EXISTING ORDER SYSTEM

If the existing system already has an order system, this can be connected to *Efficiency Manager.* This avoids users doing double work inserting the orders manually into the *Reporting Client*. The order system in the reporting is replaced by a component that connects to the existing database to get information about the orders. This does not only save time, it also helps to avoid the possibility for the operators to incorrectly declare the order.

3.8.2 DEFINITION OF BREAKS

The operator has two choices when a stop occurs in the production, either to directly define a reason for the breakdown or to fix the problem and restart the production before defining the reason for the stop. If the stop in production occurs because of a scheduled break there is often a start-up time after the break. This time generally includes adjustments, repairs and machine start-up time. The extra activities after a break should not be included into the breaks as they are waste in the production. Furthermore, depending on the production organization and how flexible the breaks are, extra break time can be categorized as a production breakdown.

To address the problems with extra activities connected to breaks and extra break time, two approaches are defined.

The first approach is intended for an organization with flexible break times. The breaks are all logged but all break time is classified as a planned stop, i.e. extra break time is categorized as planned stop. The break starts when the operators stop the machines and go for break. When they come back to the station they define the breakdown as a break. If the machine is still not producing, the system will ask for a new breakdown reason. The operator starts the machines or performs other activities, such as adjustment or a set-up. The activities after the break will then be given a reason and are logged as an extra breakdown.

The second approach is intended for a strict organization with low flexibility in terms of breaks. The idea is to define the start time and length of the breaks in advance. If the machine is not running before the break is over, the machine is defined to have a breakdown. If the stop depends on a late arrived operator the breakdown can be defined as extra break time.

The downside with the second approach is the problem to catch a shorter break with following adjustment. Activities done during the stated break time will be defined as breaks. Another downside is the big administration needed to define the different break times. The first approach does not require any extra administration. The downside with the first approach is the possibility to forget to define the break before starting with other activities. If the production starts before the operator has defined the break, the breakdown is automatically ended, and only waiting to be defined.

3.8.3 *Forced Breakdown Definition*

If operators do not see the point in using *Efficiency Manager*, the motivation to use it correctly will be low. If they start to ignore to report breakdowns as they appear, the breakdowns are going to be put in a queue. A big queue is a risk of incorrectly registered breakdown reasons. The add-on "*Forced Breakdown Definition*" is then, by using an output signal, restricting the physical process to start. The downside is how the focus is moved from restoration of the process to instead focus on the touch screen and register the stop. Other incentives should be considered as bonus goals for the measuring.

3.8.4 *Short Stoppage Registrations*

Short stoppage registration could be enabled to use the capabilities of the system to register all production disturbances. Short stoppages are normally hidden in low performance values. The short stop registration makes it possible to define and log the stoppages in the same way as *Efficiency Manager* log long stoppages. The add-on will still reduce performance as short stoppage occurs but all stops are possible to follow up. If this add-on is implemented the *Process Interface* will record much shorter stoppages. The *Reporting Client* defines if the stop will be logged as a short stop or a breakdown.

There can be problems in some systems with a low or variable production rate to accurately recognize when there is a stoppage. For some systems, the system could use some internal signals in the machine to quickly recognize a short stoppage.

Many registrations results in a more work for the operators. The gained information must be evaluated against the increased effort needed from the operators. In some cases it could be implemented temporarily to be deactivated later when some information has been collected. Short stoppages are far more common than breakdowns. A shorter time is needed to be able to evaluate the currently biggest problem in the system.

4 IMPLEMENTATION

The implementation phase acts both as a test of the proposed specification and a platform for the final system. The development is executed in parallel with the specification. Once the specification is tested in the implementation, it is revised with both minor modifications and bigger design changes. It is not always obvious how a costumer want a feature implemented; it leads to multiple alternatives and add-ons to the standard procedure. The main issue is to try to keep the software as light and adaptive as possible.

Three parts are implemented, the server with a SQL-database and a windows service, the *Reporting Client* with base functionality and a Report Generator. From the beginning only the server and Report Client were intended to be developed, but the project had good speed and some extra time was found for the *Report Generator*. The *Report Generator* gives extra value to the project and is used to shows how the stored information is intended to be used.

The first step in the development cycle was to create a SQL-database that stores the required data according to the specification. The database has been revised a couple of times when new data was needed because of revisions of the specification. However, the major structure of the database has been stable.

The second step was to develop the *Reporting Client*. In parallel with the development of the *Reporting Client*, the windows service was developed as new information was needed. The service grew constantly throughout the project. The service could be developed to provide the exact information needed within the client. No time was spent on revisions in the service when the feature is working. Thus, performance is not optimized in the server.

4.1 SERVER IMPLEMENTATION

The implementation is done very much according to the specification. The database is built according to the ER-diagram and is accessed through a windows service. The service connects to the database using LINQ to SQL. LINQ to SQL is a Microsoft programming technique that allows developers to work with SQL-tables as objects in runtime. Lots of the configurations of the connection between the application and the database are done automatically. The configurations to the connections are stored in the application configuration file and can be changed depending on the system the application is implemented in. (Microsoft, 2010)

An important design choice is made in the implementation of the service; lots of processing is done both in the SQL-database and in the service on the server. Too much processing in the server limits the scalability of system. Limited processing in the server makes the system easier to expand. If the current implementation experiences problems with performance, there are a lot of optimizations to be done with the service before restructuring the work done on the server-side. The positive part of server-side work is the ability to use simpler clients. Too much work for the client results in more expensive clients; which depending on the number of clients has great economic drawbacks.

The server was during the implementation and testing, run on the same computer as the clients. During tests with 1 second updating frequency, which can be considered as a lot faster than needed, and one *Reporting Client,* the processor for an Intel Centrino Duo 2.0 GHz was occupied by the service to 20 %. Much can easily be done with performance such as dropping the updating frequency. For some heavy calculations the frequency can be dropped by over 15 times the tested frequency without affecting the experience for the users. The network traffic in the current implementation is considered very low. In the test with 1 second updating frequency, less than 15 values of maximum 64 bit are transferred every second.

4.2 Reporting Client

The implementation of the *Reporting Client* is intended to show an example of how the vital touch screen client is intended to operate. It implements all functions stated in the specification, Section 3.7.

4.2.1 *Graphical Interface*

The GUI for the client consists of three major parts. The bottom left covers the different page categories. It covers the different monitored production parts and the administration page. To the top left the pages for each category is displayed. When a page is opened by tapping, it is displayed in the major view in the center of the client. All opened pages have their own tab in the top main window. Each tab category has its own tab set. This is displayed in Figure 8, Figure 9, Figure 14 and Figure 15.

The implementation of the *Reporting Client* is much about the graphics. The base GUI was derived from a previous case. The GUI was customized to fit the requirements of *Efficiency Manager* but most of the implementation could be reused. The base GUI development resulted not only in a good graphical environment for *Efficiency Manager*; it resulted also in a GUI template to use as a base for general graphical software developments. Using a base GUI template speeds up the development of general GUI software.

4.2.2 *Administration*

The start-up and installation phase is simple. After the connection to the server is configured, the user can supervise every predefined production part. From a system tree retrieved from the server, the users choose which production part to supervise. The choice is only done each time the program is started, and it is not necessary to shut down the client during the night. Figure 8 shows how the client is connected to a production part.

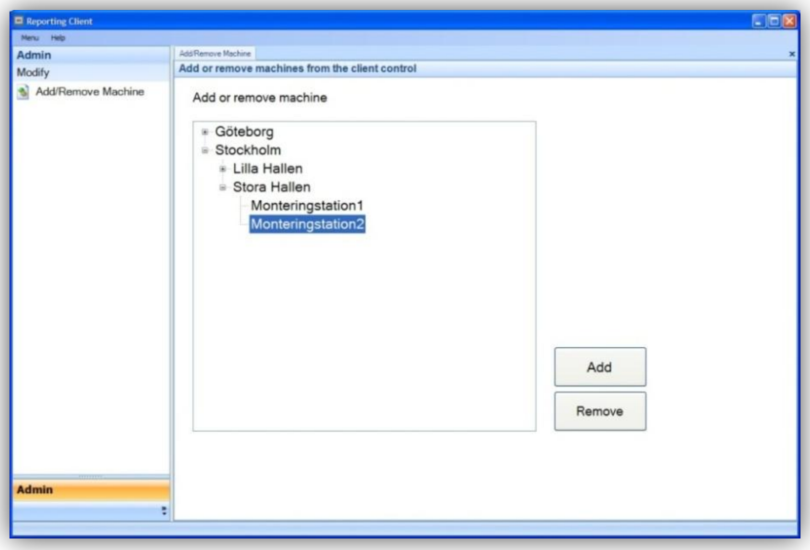


Figure 8 The client uses a system tree to define which production part to supervise.

For each production part the client is monitoring, one set of tabs for is created. In Figure 9 one production part is supervised. To the left in the window the available views for each production part are visible. The links includes access to statistics of the production part, changes to previous manual choices and back to the overview showed to the left in Figure 9.

4.2.3 *Overview*

The overview, see Figure 9, contains the information and the control functions needed for the production part. The operator defines the next product and the amount of products to produce. The amount of products to produce becomes the high limit for the progress bar shown in the GUI. When the production starts and the system begin to register produced products, they are shown in the progress bar. The progress bar shows how much of the current order that has been produced. The operator is able to follow the progress of produced products for the order in the GUI.

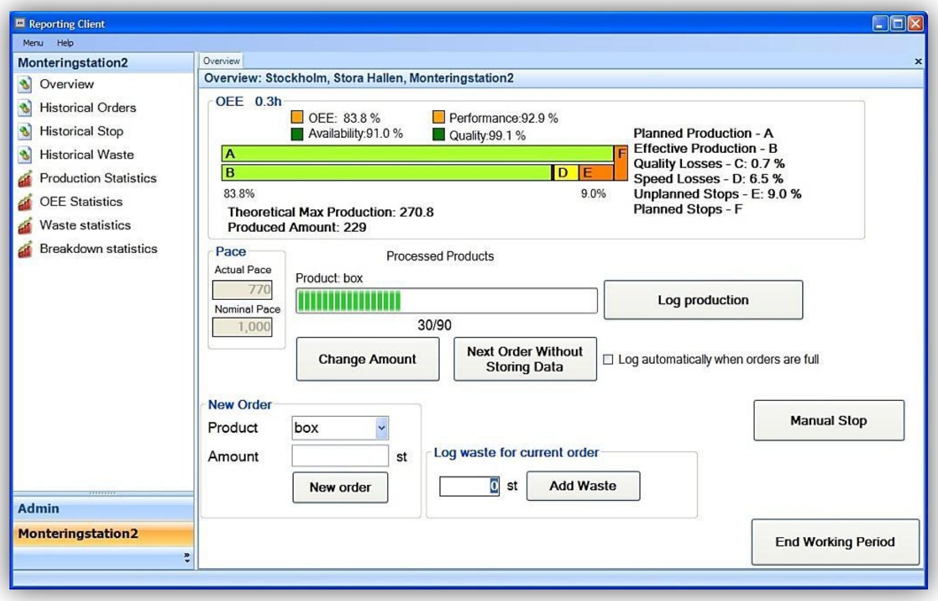


Figure 9 A standard overview for one production part in the Reporting Client.

4.2.4 *Waste Registration*

When products are wasted it is reported in the overview tab, in the field "Log waste for current order". It is also possible to register waste for an old order, see Section 4.2.8. The amount of wasted products is entered with the help of a numerical touch input panel, see Figure 10. The touch input panel appears when the operators push the amount text field. The same touch input is used for definition of the amount of products to produce in the overview and every value input throughout the project.

Figure 10 Numerical input panel for the touch screen.

A waste registration requires a reason. The choice of the reason is made through a popup. The popup is shown in Figure 11 (The same GUI for the popup is used for breakdown reason registration).

Figure 11 Registration of waste reason (the reasons is in Swedish defined earlier by the administrator).

4.2.5 Breakdowns

If a breakdown is registered by the client the reason has to be reported. The client opens a popup that has to be used before anything else is done on the client screen. This forces the operator to register every breakdown before he defines a new order. The total time for breakdowns to be queued until the breakdowns are reported is then limited to the time for one order. The longer time since the breakdown occurred and all the more breakdowns to define, it is harder to remember the real reason for the stop. The time from restarting the process after a breakdown until defining of the breakdown reason in the client must be minimized. Other possibilities are stated in Section 3.8.2 and 3.8.3.

When the operator switch order the order log activity is finished. Changing the order is done by clicking the "log order" button or by choosing "changeover" in the breakdown popup. The orders define the time for available production time. As long as the breakdown reason "no new order", see Figure 12, is not chosen an order is operating. In the start of the day when the operator has not yet defined the order, the client is not displaying the produced products but it is still recording the start time and products produced. This makes it possible for the operator to focus his work on the process before defining the order. But when the "No new order" has been defined the operator has to restart the recording by defining a new order.

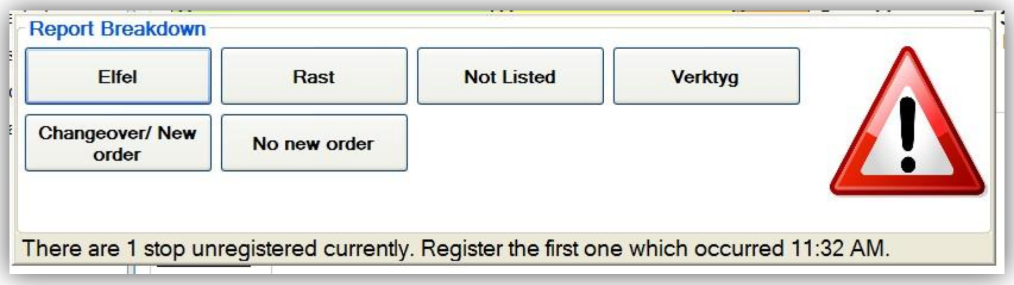


Figure 12 Definition of breakdown reasons (the reasons is in Swedish defined earlier by the administrator).

4.2.6 OEE Graph

Early stages of the implementation focused on how to present the key values for the operator in a good way. The OEE graph is a typical example of how it is possible to present OEE key values. It is important that the operators are able to see the current performance of the process. The visual feedback helps the operators to know how the process performs and where the problems in the process occur. Liker (2004) stated that visual controls should be used to visualize problems. When problems are visualized they can be resolved earlier and with less effort. If an operator sees a dropping performance he can try to speed up the operation. Without any visualization of the performance the operator would probably keep the operation in the same speed.

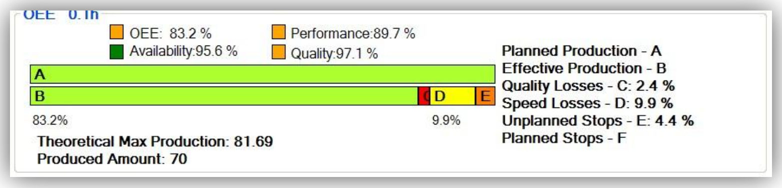


Figure 13 The presentation of the current stat for OEE-values in the last 8 h.

The idea about the OEE graph is to graphically show the operator what is included in the OEE-value and how the different production losses are affecting it. When the operator is able to visualize how his own and the stations performance affect the graphics, this supports the operator to detect and avoid common problems. Liker (2004) discusses the importance to know how the operation is performing. If the performance is not visualized nothing is likely to be improved.

4.2.7 Statistic Pages

On the clients there are a couple of predefined pages to show basic statistics of the production part. The purpose of the statistical pages is mainly to enforce the interest in the performance of the production. Feedback to the operators is important and makes it possible to see the effects of their actions. In the implemented *Reporting Client* four reports can be generated.

- Produced amount for each day during last week, see Figure 14.
- OEE-values for each day during last week.
- Amount wasted products each day during last week.
- Total breakdown time for each day during last week.

All the statistical pages are created using the same templates as the *Report Generator* do, see Section 4.3. Hence, the reports are changeable to any report that can be created by Report Generator. In the current implementation, adding or changing the report cannot be done without configure the client programmatically, but does not require much effort.

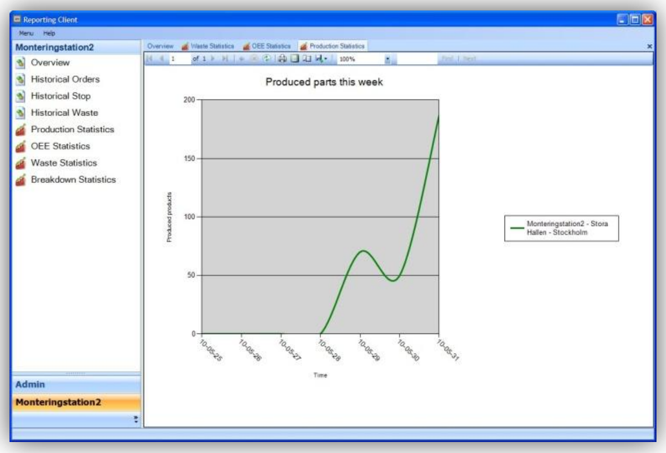


Figure 14 The example shows one of the statistical reports.

4.2.8 *MANUAL CHANGES*

All choices made by the operator can by mistake be declared wrong. Hence, it is possible to change all manual declarations in three tabs, *Historical Orders*, *Historical Stops* and *Historical Wastes*. In Figure 15 one of the three tabs are showed. All modifications done in the tabs are logged in a change log together with user name, date and information about what is changed and the new values. The change log is implemented with information enough to restore the manual changes. Four properties can be added or changed in retrospect.

- In Historical Orders the operator can add waste to a previously registered order.
- In Historical Waste the operator can change reason for previously registered wastes.
- In Historical Waste the operator can change amount for a previously registered waste.
- In Historical Stops the operator can change the reason for previously registered breakdowns.

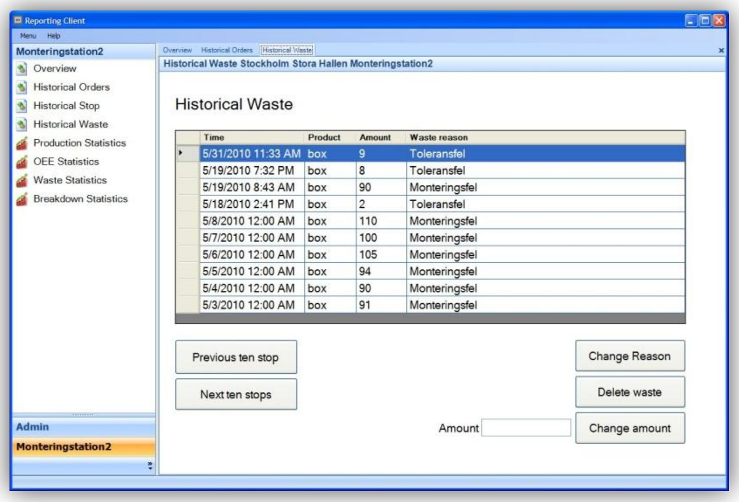


Figure 15 An example of how the manual changes tabs are implemented.

4.2.9 *MANUAL STOPS*

When maintenance is done on a Process Interface, there is a possibility that the control system incorrectly indicates that something has been produced, e.g. tuning tests after a set-up requires that the system is operating but no real products are produced. It is then possible to either lock the current stop or create a manual stop, i.e. produced products are ignored and the stop remains. The manual stop simulates a breakdown in the *Efficiency Manager* and remains until it is unlocked. In Figure 9 the button manual stop is shown. When a stop is active it changes the button “Manual Stop” to “Lock Stop”.

4.3 Report Generator

To show the value in the information collected by the system, *Report Generator* is developed. It was developed as a website using ASP.Net. From the *Report Generator* it is possible to retrieve statistical information about any time in history of *Efficiency Manager*. An example is displayed in Figure 16 showing how different choices are intuitive selected on the website. The resulted report in the bottom of Figure 16 can easily be exported to pdf, doc or xls format.

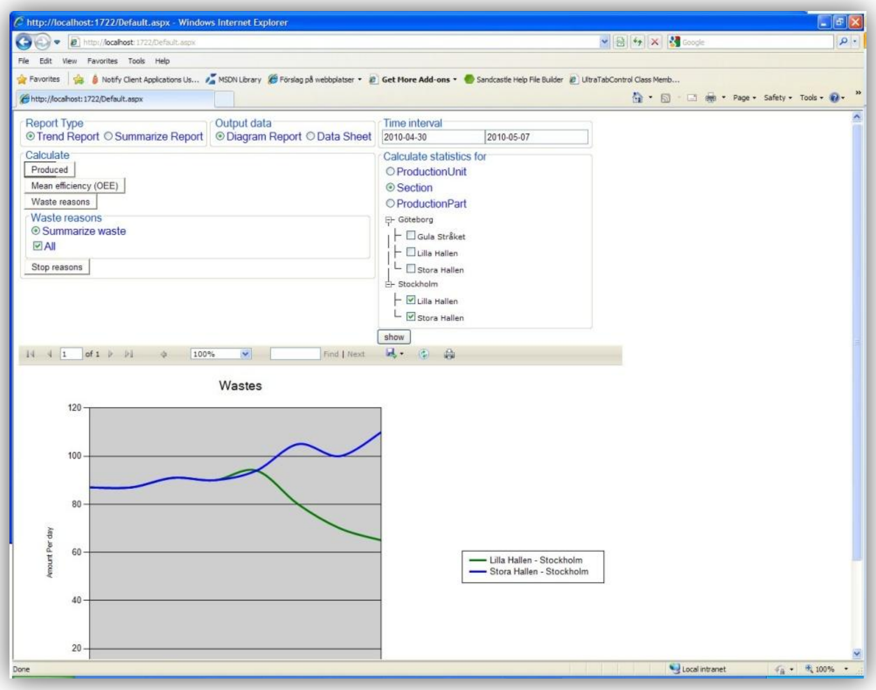


Figure 16 Generated diagram of amount wasted products, comparing two sections.

The framework for the *Report Generator* is intended to be used in other follow-up applications. If the customer for example wants a big screen in the production with updated OEE values it is easy to develop a website or standalone application using the framework provided.

The *Report Generator* uses Reporting Systems from Microsoft. Reporting Systems can be tightly integrated with a SQL-server. However, in this implementation the report is generated on the client-side using the Windows service to retrieve information. This approach has some performance issues compare to generating a report tightly coupled with the SQL-server. However, reports are not generated continuously; therefore the performance of this feature is not as important as in the *Reporting Client*. If the users experience performance issues with the approach, another method could be tested. The benefit with the current implementation is an easily expandable platform with lots of possibilities.

The current implementation uses many calls to the server. Depending on how many production parts and the length of the time-span of the examined trend diagrams, the number of calls will expand. A big performance improvement change for performance is to exchange the many calls with fewer big SQL-queries. The SQL-queries should still be executed in the server, but the service will use better and more powerful SQL-queries.

4.4 SYSTEM TESTING

Tests of the system, without a fully developed Process Interface, are performed using a *Process Interface* Simulator. The function of a *Process Interface* Simulator is to retrieve signals from, in this case, a simulated machine, see Section 3.6. The software makes it possible to graphically simulate the signals from a real Process Interface. The simulator uses the same methods as in the service used by a real Process Interface. The simulation software is used continuously throughout the project and makes it possible to simulate the dynamics in a system, including some issues with time dependencies.

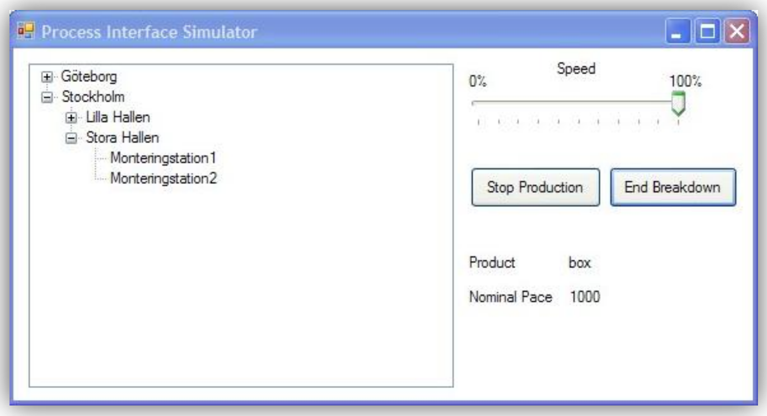


Figure 17 The Process Interface Simulator has the same behavoir as a real Process Interface.

5 DISCUSSION

A proposal on how to implement an OEE measurement system and its required functions is specified in this report and presented as a specification in Appendix A. The specification is developed by support of discussions with people working with similar systems and specifications. The general specification is created on empirical knowledge as well as tests in an implementation. Hence, the specification result can be used as a base for a real implementation. A company, who wants to measure OEE continuously or use the other tracking functions, will have use of the specified system.

Different companies will have special prerequisites in their production system. However, the report shows that the system can handle many requisites and make use of an integration of other production systems functionality. The extensions that have been discussed in this report make use of existing systems and expand the current functionality. For example, the system can be used to supervise production environment factors; the extension is implemented by storing different measures together with the breakdown events, waste events and continuously with the order.

The components required to calculate the OEE-value seems to cover many important aspects of production performance. The components can be used to give the user additional detailed information of actual problems. There is a need to complement the breakdowns and waste with more detailed information about the breakdown reason in order to take benefit of the data. If only OEE values were wanted, the stop only has to be categorized as planned or non-planned. To make such an option for a production stop the operator has to be trained. The current solution gives the operator a more hands on decision. An option between a predefined electrical problem and a break is easier to make than planned or not-planned.

The declaration of reason for waste is not needed at all for the OEE-calculation, but it gives lots of important information about problems in production. The additional information gives more value to the costumer than OEE-values. OEE-values can be used to see trends and to visualize operations with production problems. The big values for the costumer are the information about the reason to the production losses.

The project focuses on the functions and the possibilities, not on the graphical presentation. However, some issues regarding user operability and presentation are taken into concern. The GUI in the *Reporting Client* is adapted for touch functionality. This makes the GUI more intuitive and easily accessible. Depending on the touch screen hardware it could be possible to operate with gloves. The focus in the graphical presentation of the GUI is to give the user fast and accurate information, but also to have an intuitive and easy input interface.

In the implementation three important graphical components shows the current state of the production. These components are shown in Figure 9 and in Figure 12 in Chapter 4, and are discussed next.

The OEE-value is presented in a graph including the three components of the OEE-value. The values give the operator an understanding of the measured parameters. The operator is able to understand what affects the OEE-value. A bad performance by the operator or the machine affects the graph.

The progress bar shows how much of the current order that has been produced. This function makes it easier to prepare set-up in time.

The third important graphical indicator is a stop reason definer. To encourage the operators to categorize the stop, the registration is as fast as possible without losing accuracy in the categorization. The stop popup blocks everything else in the GUI. Before the operator is able to define the next order, the stop has to be defined and the popup will close.

For the system to be accepted by a customer, the design and look and feel in the GUI is of great importance. The discussions throughout the project were intended to focus on the functionality not the graphics. Nevertheless, the discussions contained lots of remarks for the graphical presentation. It is what you see that is easy to discuss. The same results are gained by visualizing the problems in production. If the operators and production personal are able to see and track the performance and problems in the production, the issues contributing to a lower efficiency will be revealed.

Another important issue is the ability of a fast and easy use of the system. If the system hinders the operators in their daily work, this will decrease performance. By requiring as few clicks as possible and keep the functions as simple as possible, the effort for the usage of the system is low. The development of the system has considered not forcing the operators to handle the system before dealing with the production. First priority is to produce, second is to retrieve information of the process. This can naturally, as discussed in Section 3.8.3, lead to ignorance of reporting issues. Therefore it is important to tell the operators how the system improves the production and to give them feedback on good work.

The *Report Generator* was developed during extra time when the development of Report Client and the server was completed. The *Report Generator* is a tool used to closer connect the production and the engineering departments. Through an internal webpage it is possible to retrieve information about the current or historical situation in the processes. The simplicity to retrieve specialized reports makes use of the retrieved data in more situations, e.g. instead of starting a new study of how the new product design affected the production, reports can directly be retrieved after the first day of production. The simplicity is the greatest value for the customer. The *Report Generator* is only one example on how to gain advantages from the collected information. It could be extended and give more and customized reports to the customer, e.g. real time large displays to visualize current state in the production.

The simulation of the implementation, with the *Process Interface* Simulator show that the concept for the specification works. A roll out in a real internal network would probably not yield in any major problems. The HTTP protocol used is often let through firewalls. Furthermore, the solution only consists of one server; the clients do never act as servers. Consequently, adding a new client does not require the administrators to add addresses for the clients in the server, instead all the clients are configured with the same server address.

5.1 Scalability and Performance

The amount of processing performed in the server both in the SQL-database and in the Service is important restrictions for scalability. In the present implementation clients do not cache any data from the SQL-database. When a client needs a value, it connects to the service, which performs the calculations and queries. Lots of performance can easily be gained by caching data and use versioning. The scalability of the current implementation is limited because of the refresh rate in the *Reporting Clients*. Moreover, lots of work is now executed by the server. There is much to gain in server performance to move work to the clients. By caching log information required for the OEE calculation, the refresh from the client would only check and retrieve the updates. It would benefit a lot to server performance since that is the most common activity for the service.

The implementation of the service is not focused on performance. Many of the functions included in the service have big improvement potential. By minimizing the iterations in the service and do smarter and bigger queries to the SQL-database, a lot of performance can be gained. If this system is released to the market the server performance is an important issue.

5.2 CALLBACK APPROACH

A different approach for the communication is to focus more on event based updates instead of the current poll based approach. An event based update approach allows the server to make callbacks to the clients when a value important for the client is changed. The approach is especially good for seldom changed variables, which are important for fast retrieval for the clients. The approach in the current implementation is that the clients poll for every needed variable. The effect is a highly overused network. However, from discussions with experienced personnel the overuse was considered a small problem. Experience tells that there could be firewall and implementation problems with a callback approach.

6 CONCLUSIONS

The system specified in the report and provided in Appendix A, gives an example of how a system for OEE measurement could be implemented. The system is flexible to install and expand. It is easy to use and minimizes the influences on production.

The implementation shows that the functionality of the system works in a simulated environment.

7 FUTURE WORK

To implement the system in a full scale production environment some work remains. The administration client, which mainly is a webpage, is not implemented in any matter. The implementation of the webpage is uncomplicated and direct. The information to be put into the database from the webpage is a matter of direct mapping. After the admin page is built, a couple of tests, including performance tests, in a full scale simulated production must be conducted. If there are performances issues, there are lots to do with the server, read the discussion in Chapter 5.

After good performance results from simulations, the system can be implemented and tested in the intended hardware for the system. Hardware development includes development and testing of the Process Interfaces to measure signals from the production, and development of the *Reporting Client* in a touch screen. When the systems are ready, field tests can be conducted.

If it is too much investment and the project comes with a too big risk, recommendation is given AcobiaFLUX to continue the development within the scope of a new master thesis. The tests and hardware implementation could be a good training for a product development student.

The value gained from this system is worth a lot and the product has big potential to be a success for AcobiaFLUX.

7.1 FLOW ANALYSIS

Efficiency manager provides statistics about how well exploited a production process is. To get more information about the interaction in the total production flow, a flow analysis must be performed. A case study made by Alin et al. (2009) shows a performance increase with more set-ups. Hence, more set-ups results in a more flexible production. Therefore, the total system became more efficient. Nevertheless, the efficiency decreased for some individual processes. A flow analysis needs lot of information about cycle time breakdowns and quality rate. This information can be retrieved from data gathered from *Efficiency Manager*. A future work could focus on how to support flow analyses to investigate the total production flow with information from individual processes. Hence, investigate how to merge data gathered from *Efficiency Manager* with a flow analysis tool. As the case showed, high individual efficiency in each process could be a contra productive goal if the operations depend on each other.

8 BIBLIOGRAPHY

AcobiaFLUX, 2010. *AX CARAT - Produktblad*. [Online] Göteborg: AcobiaFlux (RevB) Available at: http://www.acobiaflux.se/produkter/ax-carat.php [Accessed 10 May 2010].

Alin, D., Andersson, J., Andersson, M., Isaksson, A., Skoogh, A. and Helander, E. (2009) 'Examining the Relation Between EPEI-Time and Productivity Using Discete Event Simulation', Proceedings of the 2009 Swedish Production Symposium, Gothenburg, 295-301.

Jonsson, P. and Lesshammar, M. (1999) 'Evaluation and improvement of manufacturing', *International Journal of Operations & Production Management*, vol. 19, no. 1, pp. 55-78.

Liker, J.K. (2004) *The Toyota way: 14 management principles from the world's greatest manufacturer*, New York: McGraw-Hill, ISBN: 0071435638.

Microsoft (2010) *MSDN Libary*, [Online], Available: http://msdn.microsoft.com/en-us/library/ [03 May 2010].

Muthiah, K.M.N. and Huang, S.H. (2007) 'Overall throughput effectiveness (OTE) metric for factory-level', *International Journal of Production Research*, vol. 45, no. 20, October, pp. 4753-4769.

Nakajima, S. (1988) *An Introduction to TPM*, Portland: Productivity Press.

Nakajima, S. (1994) *TPM Development Program*, Portland: Productivity Press, ISBN 0-915299-37-2.

Scott, D. and Pisa, R. (1998) 'Can overall factory effectiveness prolong Moore`s Law', *Solid State Technology*, March, p. 75–82.

APPENDIX A

The following appendix is written in Swedish.

Innehåll

1 Inledning

AcobiaFLUX säljer in ett koncept kallat Efficiency Manager. Konceptet riktar sig till kunder som vill kunna spåra problem i sina processer. Med spårbarhet får kunderna möjlighet att fokusera sina förbättringsinsatser på processer där de största problemen finns. Idén är att låta systemet avläsa produktionshastigheten och jämföra den med processens teoretiska hastighet. Om systemet ser att processen har stannat, ger operatören information om orsaken till stoppet. Konceptet innebär också att det blir möjligt att visa effektivitetstal och kvalitetskvoter i realtid för de olika processerna. Det ger bättre feedback till operatörer och personal på managementnivå för styrning och planering.

Emballator Mellerud Plast har idag ett system för produktuppföljning och störningshantering. Det systemet innehåller en hel del felaktigheter som inte kommer att åtgärdas. Kunden vill istället skapa ett nytt likvärdigt system som tillgodoser de krav som ställs och få en fortlöpande utveckling på systemet. (Jönsson, 2009)

Milko har idag ingen spårbarhet på var eller när problem uppstår i produktionen. De vill kunna se vilka produktionsavsnitt som är de största problemen, för att ha möjlighet fokusera underhåll och förbättringsprocesser. (Esping, 2009)

1.1 Konceptbeskrivning

Grundkonceptet är att med hjälp av information om den aktuella produktionstakten, logga produktion och stopp i produktionen, och med hjälp av informationen beräkna effektiviteten.

Vid produktionsavsnitt i fabriker som implementerar konceptet skall touchskärmar finnas. På touchskärmen väljer operatören vilken produkt som skall produceras. Systemet kan då jämföra aktuell takt med börvärdet för produktionsavsnittet. Om processen stoppas under produktion av någon orsak visas det på skärmen. Vid stopp skall operatören välja med maximalt två klickningar på skärmen vad som orsakat stoppet. Två klickningar innebär ett snabbt förfarande men också möjlighet till många och speciella orsaker. Stoppet loggas och används för beräkningar och statistik för processen. Med hjälp av information om produktionstakt, stopptider, planerad produktionstid och antalet kasserade och godkända produkter kan effektivitetstal beräknas, se kapitel 3. Effektivitetstalen visas direkt på skärmen och ger snabb feedback för hur produktionsavsnittet presterar till operatören.

I efterhand är det möjligt att ta fram statistik för delar i fabriken. Statistiken ger trender och jämförelser och ger förutsättningar för företaget att hitta problem samt se effekter av förbättringar i processen.

1.2 Syfte

Flera parallella projekteringar på liknande problem hos kunder finns hos AcobiaFLUX. AcobiaFLUX vill bygga en plattform som tillgodoser kundernas behov. Syftet med detta dokumentet är att sammanställa funktioner som skall finnas i ett generellt system som klarar olika kunders behov. Implementering enligt specifikationen skall speciellt kunna appliceras på Emballator och exemplet från Milko. (Esping, 2009)(Jönsson, 2009)

1.3 Grundkrav

Systemet skall vara modulbyggt, enkelt att vidareutveckla och modifiera för specifika behov. Det ställer krav på en väldokumenterad och strukturerad design vars delar är ordentligt testade.

Systemet skall ta så lite tid av operatören som möjligt. Om tiden för initiering och inmatning i systemet blir en betydande del av operatörens arbete, bör tillägg för automatisk registrering införas i större utsträckning, se avsnitt 9.4.

1.4 Definitioner

Allmänna definitioner för dokumentet.

1.4.1 Produktionsavsnitt

Den delen som beräkningar och mätningar utförs på kallas produktionsavsnitt. Det kan vara en maskin eller station men även flera stationer eller maskiner efter varandra, till exempel en linje. Från ett produktionsavsnitt skall systemet ha möjlighet att få information om produktionstakten. Kvaliteten för produktionsavsnittet skall kunna kontrolleras. Produktionsavsnittet skall processa en produkt eller produkttyp med samma börvärde för produktionstakten åt gången.

1.4.2 Avdelning

Grupperingar av produktionsavsnitt som har samhörighet med varandra benämns avdelningar. Samhörigheten kan vara i form av produktionslayouten eller produktionsenheternas funktion. Avdelningarna används för jämförande i statistiken och för att få en logisk uppbyggnad av fabriken i ett trädsystem.

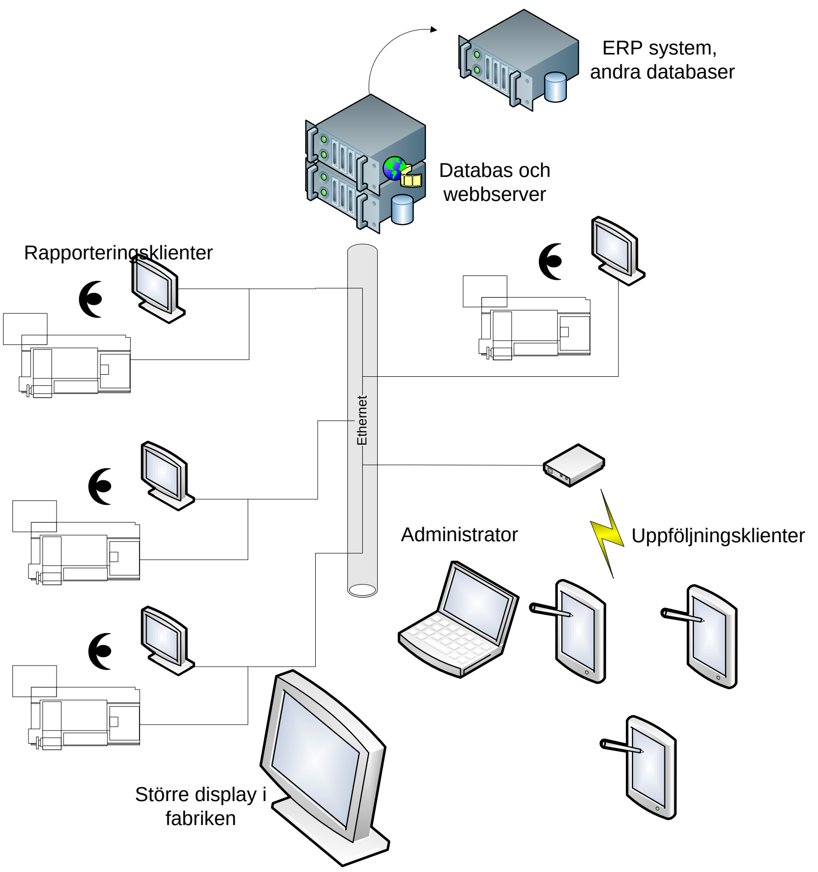
1.4.3 Produktionsenheter

Gruppering av avdelningar benämns produktionsenhet. De används på samma sätt som avdelningar men är en nivå över avdelningar i trädstrukturen. Produktionsenheter kan vara olika fabriker eller större enheter i en fabrik innom företaget.

2 Systemöversikt

Systemet består av en Server, Rapporteringsklienter, Administrationsklienter, Uppföljningsklienter och ett systemspecifikt processgränssnitt för avläsning av status för produktionsavsnitt till Servern. Eventuellt kan systemet kopplas ihop med ett existerande system. Figur 1 visar hur systemet kan vara uppbyggt.

Systemet har stöd för användarsäkerhet i olika nivåer. Information kan döljas eller visas för olika grupper.



Figur 1. Konceptbild av systemet

2.1 Server

Systemets hjärta är en Server med en databas. Alla klienter i systemet skriver och läser information från servern. Här finns loggar av all produktionsinformation inklusive kassation och produktionsstopp.

Servern har även en webbserver som skapar rapporter om produktionen. Statistik över produktionen och dess effektivitet kan nås direkt via ett webbgränssnitt.

Vidare information se kapitel 4.

2.2 Administrationsklient

Systemet administreras från en webbläsare. Administratören kan från webbläsaren konfigurera systemet och skapa produktionsenheter, avdelningar, produktionsavsnitt, produkter, kassationsorsaker och stopporsaker. Klientapplikationen är designad i ett enkelt användargränssnitt, med åtanke att företaget själva skall kunna modifiera systemet utan extern konsult.

Klienten beskrivs närmare i kapitel 5.

2.3 Rapporteringsklienter

Rapporteringsklienterna ute i produktionen kan konfigureras för att övervaka ett eller flera produktionsavsnitt. I klienten ställer operatören in vilken produkt som skall produceras och hur mycket som skall produceras. Klienten hämtar information för produkten i det aktuella produktionsavsnittet från Databasen. När produktionen är igång visas processens effektivitetstal och uppmärksammar operatören om produktionen stannar. Effektivitetstalen ger en återkoppling till operatören som tydligt ser om produktionen går bra eller dåligt.

Klienten är implementerad i en IP67-klassad touchpanel. Gränssnittet är anpassat till en snabb och enkel användning för operatören. Påverkan på produktionseffektiviteten blir då låg eller obefintlig.

Funktionerna i rapporteringsklienten är modulärt uppbyggda för att anpassas till behoven på företaget. De funktioner som inte används kan plockas bort vid anpassning till företagets verksamhet, behov och system. Ett exempel är en produktionsmiljö med endast en produkt. Operatören behöver då inte välja en produkt som skall produceras utan loggning av producerad mängd sker kontinuerligt för varje produktionsavsnitt.

Om delar av informationen som skall behandlas och lagras redan hanteras av andra system skall den informationen tas tillvara på. Rapporteringsklienten kan anpassas för att eliminera dubbelarbete för operatören. Exempelvis, om ett ordersystem redan finns skall inte en order definieras på två ställen. För att ta tillvara på informationen kan det krävas vissa anpassningar av lagrad information för att passa in i systemet.

Klienten beskrivs noggrannare i kapitel 6.

2.4 Processgränssnitt

Produktionsavsnitten har olika teknisk lösning på hur produktionen kommunicerar och vad den tillhandahåller för information. Det gör att gränssnittet från process till databas skapas kundspecifikt. Kravet är att produktionsavsnittet skall kunna tillhandahålla information om produktionshastigheten. I grundsystemet accepteras hastigheten i form av en puls som motsvarar en speciell mängd produkter. Processgränssnittet uppdaterar då databasen varje gång den tar emot en puls.

Om signaler för kvalitet och automatisk kassation av produkter finns tillgängligt kan processgränssnittet använda dessa. Signalerna tas emot och översätts till loggningar av kassation. Ingen manuell registrering för de kassationsorsakerna som motsvarar signalerna behövs då.

Om produktionsavsnittet ger pulser sällan, antingen beroende på låg produktionstakt eller många produkter per puls, kommer systemet ta lång tid på sig att reagera. Om det ändå finns ett behov av precis loggning av driftstopp i ett system där pulserna kommer sällan, är det möjligt att koppla en direkt signal för driftstopp till systemet. Fördröjningar tills systemet reagerar på att produktionsavsnittet har stannat kan då elimineras.

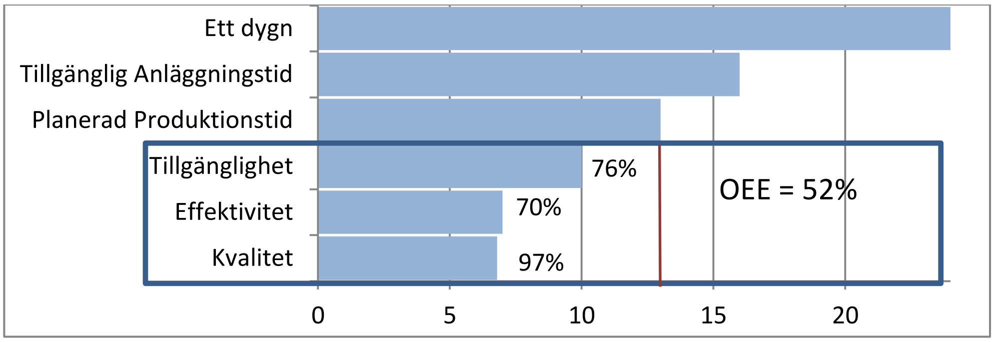
2.5 Uppföljningsklient

Via ett webbgränssnitt kan personal se produktionsdata och statistik. Rapporterna bygger på Microsoft Reporting Services vilket gör att rapporterna lätt kan skrivas ut eller sparas i format som exempelvis pdf eller Excel. Uppföljningsklienten är en del av administrationsklienten. För närmare beskrivning om uppföljningsklienten se kapitel 7.

3 OEE Beräkning

OEE (Overall Equipment Effectiveness) är ett tal för att beskriva hur stor del av effektiviteten av produktionen som är utnyttjad. Effektivitetstalet utvecklades av Nakajima, S (1988) tillsammans med hans arbete om Total Productive Maintenance (TPM) för att användas som ett mätvärde för effektiviteten i processerna. Värdet tar inte hänsyn till allt. Exempelvis tar inte OEE hänsyn till ekonomiska, sociala eller miljömässiga aspekter i processerna. Om två liknande processer får olika OEE-tal betyder inte att den med högst OEE-värde absolut är den mest lönsamma eller den bästa. Det betyder att den med högst OEE-tal utnyttjas bäst under planerad produktionstid. OEE skall främst användas till att visualisera problem, se trender i effektiviteten, följa upp hur förbättringar påverkar produktionen, och i viss mån för att jämföra processer. Jämförande kan visa på att en process har kommit längre i att förebygga problem, kan utnyttjas bättre eller ger mindre förluster på grund av problem med kvaliteten.

Tiden som OEE-talet har som grund är planerad produktionstid, se Figur 2. Planerad produktionstid är tiden operatörer och maskiner är tillgängliga för produktion minus planerade stopp. Klassifikation av planerade stopp är stopp då produktionen inte är tänkt att köras. Vanliga planerade stopp är raster, planerat underhåll och orderbrist.



Figur 2. Exempel på en OEE-kalkyl

OEE beräknas genom att multiplicera ihop tre delar:

- Availability = Tillgänglighet
- Performance = Anläggningseffektivitet
- Quality = Kvalitet

Det finns individuella skillnader i hur olika företag beräknar sina OEE-tal. Beräkningen i Efficiency Manager kan anpassas till företagets krav.

3.1 Tillgänglighet

Beräkning av tillgänglighet:

$$Tillgänglighet = \frac{Planerad\ produktionstid\ - Stopptid}{Planerad\ produktionstid}$$

Stopptid är tiden då maskinen är nere för oplanerade stopp som verktygsbyte, reparationer eller materialbrist. Även reducerad produktionshastighet på grund av uppstart av produktionsavsnittet skall enligt Jonsson & Lesshammar inkluderas i stopptider. Det är svårt att skilja på vilken komponent som är reducering av produktionshastigheten på grund av uppstart och komponenten som beror på anda faktorer. Reducering av hastigheten på grund av uppstart inkluderas därför i anläggningseffektivitet i Efficiency Manager.

Anledningen till att planerat underhåll behandlas på annat sätt än oplanerade stopp är att planerat underhåll har fördelen av att allt kan vara förberett och att tidpunkten för underhållet kan planeras. Vid ett oplanerat stopp skall reparationspersonal tillkallas, material eventuellt beställas och arbetet förberedas. Planerat underhåll är bättre än att underhålla vid haveri och påverkar därför inte effektivitetstalet negativt. Det planerade underhållet skall ändå strävas att minimeras för att öka produktionstiden. Statistik över planerade stopp tillhandahålls av systemet.

(Jonsson & Lesshammar, 1999)

3.2 Anläggningseffektivitet

Anläggningseffektivitet representerar förlustkomponenter, små stopp och reducerad produktionshastighet. Det är de komponenter som gör att den verkliga produktionshastigheten inte når upp till den teoretiska. Faktiska orsaker till detta kan vara slitna delar i maskinen, ineffektiva operatörer, små stopp som inte hinner registreras som stopp eller längre produktionstid på grund av dåligt material.

En felaktigt bestämd teoretisk hastighet påverkar OEE-talet kraftigt. Det är därför viktigt att bestämma en så korrekt teoretisk hastighet som möjligt. När man bestämmer den teoretiska hastigheten skall man tänka på att det kan vara en takt som nästan inte är praktiskt möjlig. Teoretiska hastigheten skall vara ett värde att sträva efter, inte ett värde som går att komma över. Den teoretiska hastigheten är inget värde att variera efter hur stor efterfrågan är. Då kommer OEE-talet snarare visa hur nära produktionsmålet effektiviteten är, än hur stor del av kapaciteten som är utnyttjad.

Om ett företag har en annan syn på den teoretiska hastigheten går det att justera. Men företaget skall vara medvetet om de begränsningar som blir resultatet av en justering av beräkningen.

Beräknas som:

$$Anläggningseffektivitet = \frac{Teoretisk\ Cykeltid * Producerad\ mängd}{(Produktionstid - Stopptid)}$$

Vilket kan skrivas som:

$$Anläggningseffektivitet = \frac{Verklig\ Hastighet}{Teoretisk\ Hastighet}$$

Eller:

$$Anläggningseffektivitet = \frac{Teoretisk\ Cykeltid}{Verklig\ Cykeltid}$$

(Jonsson & Lesshammar, 1999)

3.3 Kvalitet

Kvalitet definieras i OEE som kvoten mellan mängden godkända produkter och mängden processade produkter. Godkända produkter är de produkter som inte behöver någon omarbetning och är godkända från produktionsavsnittet.

Beräknas enligt följande:

$$Kvalitet = \frac{Antalet\ tillverkade\ produkter - Kasserade\ produkter}{Antalet\ tillverkade\ produkter}$$

(Jonsson & Lesshammar, 1999)

Kvalitet kan mätas genom automatiska mätningar eller okulär kontroll av operatörer. Båda metoderna stöds av systemet.

Bra litteratur för OEE-nybörjaren är Vorne Industries snabbguide. (The Fast Guide to OEE, 2008)

4 Server

Servern står till alla klienters förfogande. Den innehåller en databas som nås via en service som definierar vad som kan nås på databasen. Från servern får klienterna all information som behövs för presentationen till användarna. Servern finns tillgänglig alla tider på dygnet. Beroende på kundens specifikation kan åtkomsten begränsas.

I servern finns också en webbserver. Från en webbklient kan användarna styra och få information om produktionen. Mer om detta i kapitlen om administrationsklienten och uppföljningsklienten, kapitel 5 och 7.

4.1 Information i databasen

All statistisk information som lagras, lagras utifrån tre huvudkategorier:

4.1.1 Order

I orderkategorin sparas information som är relaterad till de producerade produkterna. Grundinformationen som krävs för att systemet skall kalkylera OEE är tider för start och stopp av en order, den producerade mängden och den teoretiska hastigheten för ordern. Systemet dokumenterar även information om vilken produkt som producerades samt den planerade produktionsmängden. Systemet kan då ge information och föra statistik på hur individuella produkter påverkar processerna och hur mycket som skulle produceras mot vad som har producerats.

4.1.2 Stopp

I kategorin stopp lagras information för varje större stopp. Beräkningen av OEE kräver tiden för stoppets början och slut och om stoppet var planerat eller inte. Systemet lagrar även information om stoppets orsak samt vilken produkt som kördes när stoppet startades. Informationen om orsaken till stoppet ger, förutom informationen om stoppet är planerat eller inte, en viktig möjlighet till uppföljning om varför produktionsenheterna stannar.

4.1.3 Kassationer

Den slutgiltiga informationen som krävs är information om kassationer. För OEE beräkningen krävs information om storleken på kassationen och en tidpunkt för att kunna relatera kassationen till en order. Information om orsaken till kassationen sparas också.

4.1.4 Extra data

Om mer information önskas sparas görs det händelsebaserat inom någon av de tre kategorierna. Om exempelvis temperaturen i maskinens lager behöver lagras, sparas ett värde för det i stopp, kassation och kanske orderkategorin. Det möjliggör statistik på lagrens temperatur för respektive situation.

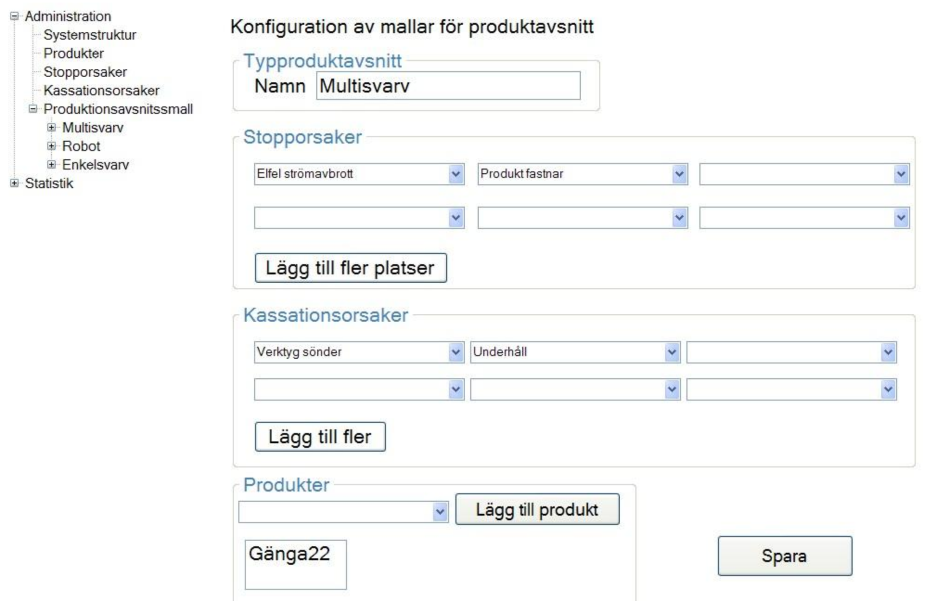
4.1.5 Loggning av ändringar i databasen

Alla ändringar som görs manuellt loggas i databasen. I loggen kan man utläsa vem som utfört ändringen, vad som ändrats, det nya värdet samt vad som fanns tidigare. Informationen gör det möjligt att återställa en ändring av en användare.

5 Administrationsklient

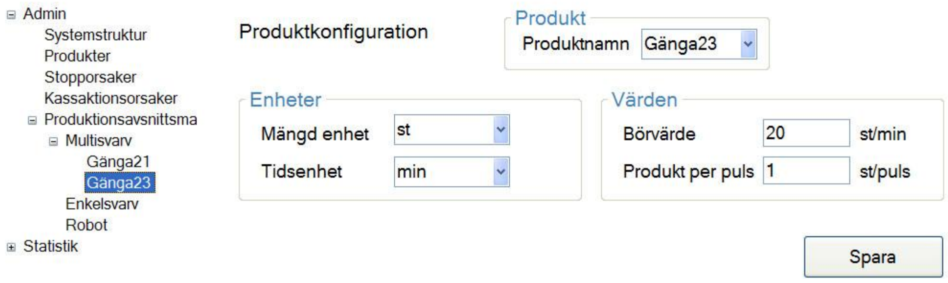
Grundidén för Administrationsklienten är att placera de delar i anläggningen som skall övervakas i en hierarki. Hierarkin definierar modulernas relationer till varandra. Längst ner i hierarkin finns produktionsenheter. De är tänkta att motsvara större grupperingar inom företaget eller olika anläggningar i företaget. Nästa nivå är avdelningar. En avdelning innebär en gruppering av produktionsavsnitt inom en produktionsenhet. Grupperingen skall representera naturliga avgränsningar i företaget, exempelvis layoutmässigt eller funktionsmässigt. I avdelningarna läggs produktionsavsnitt, som är den nivå som övervakas och mäts. Till varje produktionsavsnitt kopplas produkter och orsaker som sedan kommer utgöra informationen till rapporteringsklienterna.

Produktionsavsnitt i systemet kan definieras typmässigt. Ett företag som har flera liknande produktionsavsnitt definierar en mall för produktionsavsnitten. Mallen instansieras senare och kan konfigureras för individuella skillnader i produktionsavsnitten. I typdefinitionen anges vilka stopporsaker och kassationsorsaker som är associerade till typen, se Figur 3.



Figur 3. Definiering av produktionsavsnittsmall

Produkter associerade med mallen till produktionsavsnittet får standardvärden för börvärde och produkter per puls, se Figur 4. Produkter per puls är den mängd produkter som tillverkas av produktionsavsnittet varje gång processgränssnittet får en puls, se kapitel 2.4. Börvärde är den teoretiskt möjliga högsta takten för produktavsnittet med den produkten. Se kapitel 3.2 för kommentarer hur man sätter börvärdet (Teoretiska hastigheten).



Figur 4. En produkt kopplas till mallen

Stopporsaker och kassationsorsaker för produktionsavsnitt fördefinieras på egna flikar. När användaren skall associera stopporsaker och kassationsorsaker får användaren en lista på de fördefinierade orsakerna. Om användaren lägger till eller tar bort en orsak på en mall läggs den till eller tas bort på alla produktionsavsnitt med den mallen i anläggningen. Samma typ av orsak kan ligga på flera olika produktionsavsnitt. Det innebär att orsakerna blir underlag för statistik i hela företaget inte bara per produktionsavsnitt. I Figur 5 visas fliken ”definiera ny stopporsak”. Motsvarande flik för kassationsorsak är uppbyggd enligt samma princip.

Om fältet grupp används innebär det att de orsakerna i samma grupp grupperas tillsammans. Operatören väljer då först vilken grupp av orsaker han katalogiserar stoppet som. Nästa steg är att definiera den speciella orsaken. Operatören har då två nivåer för att välja fel. Det innebär att många orsaker kan definieras utan att antalet knappar behöver bli för många. Exempelvis kan alla typer av raster definieras i gruppen rast. När operatören väljer rast som stopporsak kommer nästa ruta upp där operatören väljer vilken typ av rast det var.

Användaren skall också definiera om stoppet är ett planerat stopp. Är stoppet ett planerat stopp påverkar inte stoppet OEE-talet.

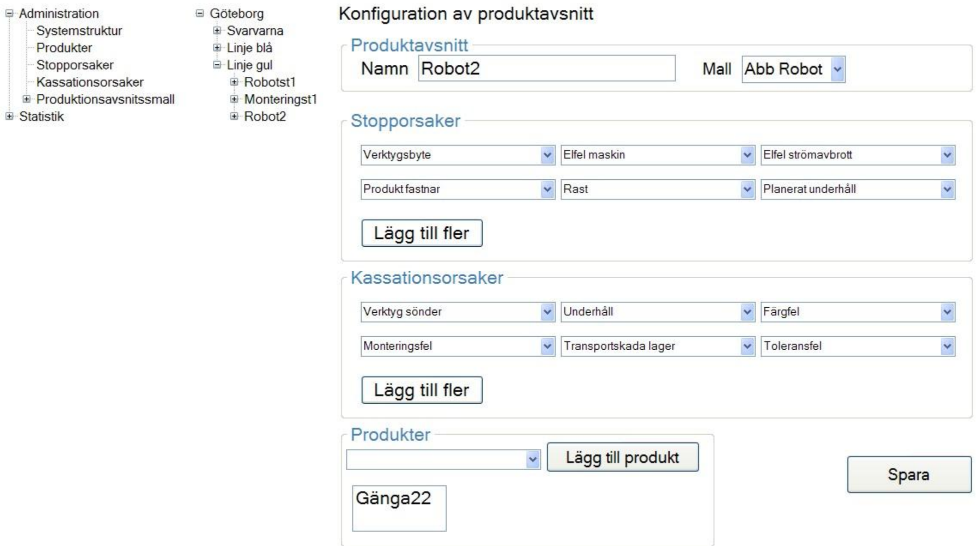
Figur 5. En ny stopporsak definieras

Produkter namnges innan de associeras till produktavsnitt eller mallar till produktavsnitt, se Figur 6. Det minskar risken för att samma produkt får olika namn. Administratören får ett extra steg att göra innan han skapar ett nytt produktnamn. När flera produktionsenheter har samma produkter associerade, kan statistik om produkterna och deras påverkan på olika processer kan genereras.

Figur 6. Ny produkt adderas till produktlistan

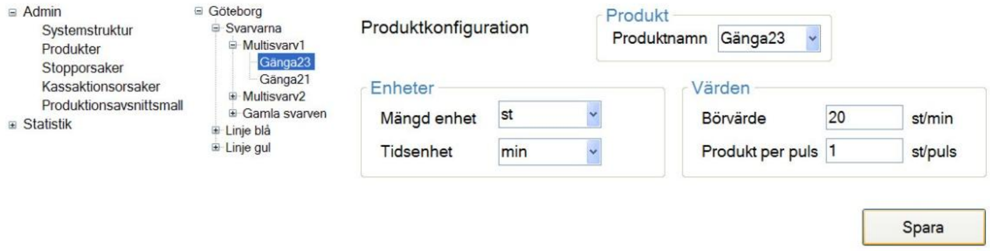
Med hjälp av alla fördefinierade typer byggs anläggningen upp. När produktionsavsnitten läggs till i anläggningen kan de modifieras med egna stopporsaker och kassationsorsaker se Figur 7. Om en produktionsavsnittsmall väljs hämtas informationen för den mallen. Informationen från mallen kan modifieras fritt. Ändras mallen senare kommer en fråga om de produktionsavsnitt som är associerade till mallens skall ställas in till mallens värde.

Några stopporsaker är speciella och finns som standard. Standardorsaker är ställ, orderbrist och egen beskrivning. Ställ har funktionen att nuvarande order loggas och efterkommande produkter läggs till nästa order. ”Orderbrist” loggar nuvarande order och loggar efterföljande tid som ”ingen produktions tid”. Egen kommentar grupperas i statistiken under övrigt och låter sedan operatören skriva in en egen orsak. Kommentarerna kan sedan analyseras för att kunna skapa nya stopporsaker för produktionsavsnittet. Övrigt definieras som oplanerat stopp.



Figur 7. Produktionsavsnitten definieras individuellt

När produktionsavsnittet är specificerat lägger administratören till möjliga produkter. Om en mall valts kommer mallens produkter att läggas till automatiskt. Produkter och deras börvärden specificeras eller modifieras individuellt, se Figur 8.

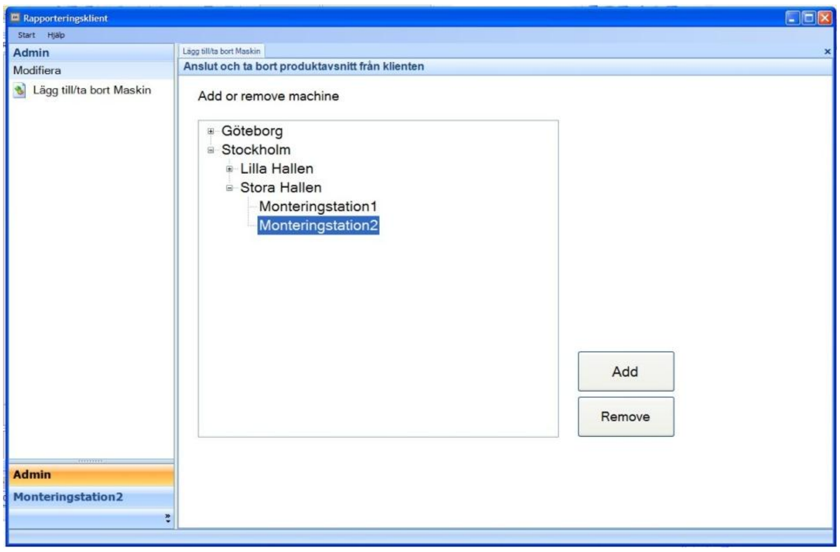


Figur 8. Två produkter har lagts till på produktavsnittet Multisvarv1

6 Rapporteringsklient

Via touchskärmarna ansluts klienterna med produktionsavsnitten i servern, se Figur 9. Informationen om produktionsavsnitten är definierade i servern sedan tidigare via administrationsklienten. Information om stopporsaker, kassationsorsaker och produkter för det valda produktionsavsnittet laddas in i rapporteringsklienten. Användargränssnittet anpassar sig enligt informationen till det valda produktionsavsnittets konfiguration och dess komponenter.

Det är även möjligt att koppla en rapporteringsklient till flera produktionsavsnitt inom samma avdelning. Under tiden som ett produktionsavsnitt övervakas grafiskt, övervakas de andra anslutna produktionsenheterna i bakgrunden. En avrapporteringsruta öppnas när ett produktionsstopp inträffar för något av de anslutna produktionsavsnitten, se Figur 12.



Figur 9. Anslutning av ett nytt produktionsavsnitt

6.1 Arbetsgång

Arbetsgången för operatören efter att klienten är konfigurerad går till som följande:

6.1.1 Start

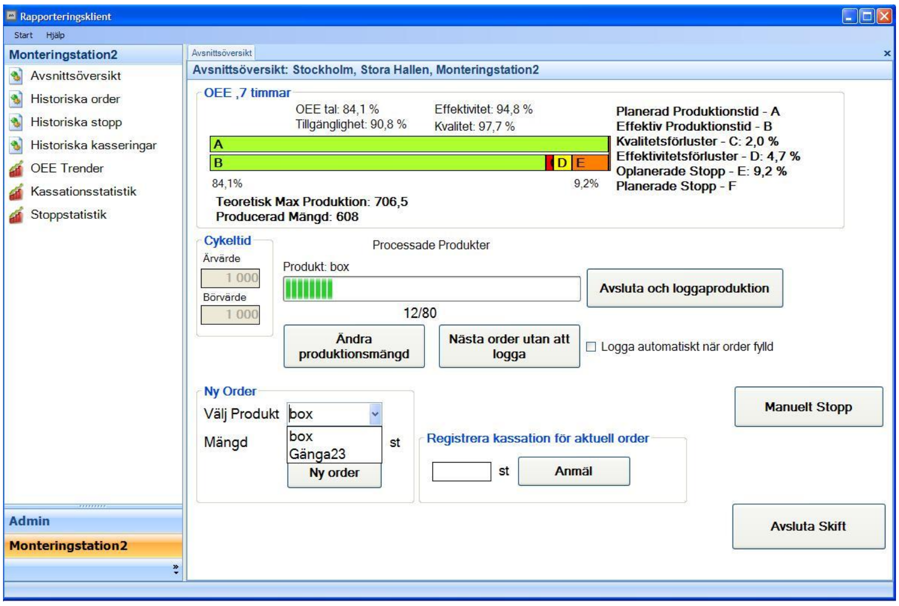
Operatören anländer till stationen och öppnar övervakningen för vald maskin via touch-skärmen. Om operatören behöver ställa om produktionen, det inte finns någon order eller dagen startas med underhåll kommer system registrera att ingen produktion är igång. En stoppregistrering kommer i så fall aktiveras. Operatören registrerar vad som orsakat stoppet, exempelvis underhåll, orderbrist eller ställ, se Kapitel 6.2.

Startandet av övervakningen bestämmer tillgänglig produktionstid. Tillgänglig produktionstid minus planerade stopp är basen för OEE-beräkningen. Tillägget för kalenderfunktion, se avsnitt 9.1, beräknar tillgänglig anläggningstid annorlunda och ingen start av övervakningen är nödvändig.

6.1.2 Orderkonfiguration

Operatören väljer en produkt som skall produceras och anger hur mycket som skall produceras, se Figur 10. Börvärde för produktionshastigheten laddas ner från databasen automatiskt. Operatören startar produktionen och information om produktionen uppdateras kontinuerligt från servern. Ett tillägg för manuell justering av börvärde finns i avsnitt 9.2.

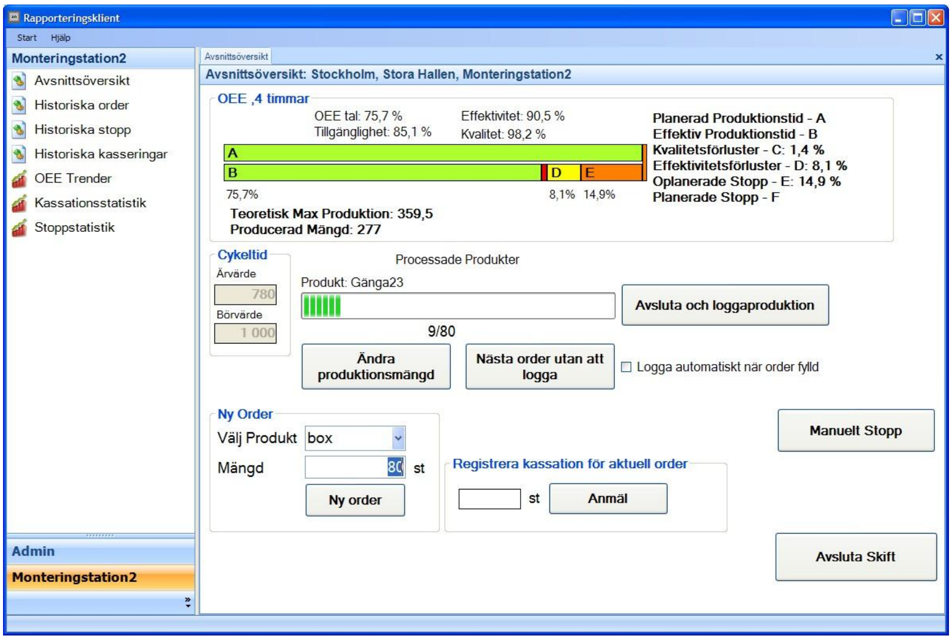
Ett alternativ till att skriva in ordern direkt i klienten, är att använda central orderplanering. De order som skall produceras läses in från företagets ordersystem. Orderna hämtas av klienten från databasen och hamnar i en kö. Operatören behöver då inte manuellt bestämma vad och hur mycket som skall produceras. Operatören byter till nästa order genom att klicka på ”starta nästa order” alternativt välja ställ/ny order vid ett aktuellt stopp.



Figur 10. **Operatören väljer bland möjliga produkter i en lista**

Klienten hämtar börvärde och aktuell takt från databasen. När produkter registreras visas det i en förloppsindikator, se Figur 11. Innan en order är klar kan nästa definieras. I den lokala orderkön visas nästa order så att operatören kan förbereda nästa ställ.

Om produktionen är kontinuerlig och inte växlar produkt kan förloppsindikatorn tas bort. I bakgrunden skapas en order automatiskt när produktionen är igång för att kunna vara till underlag för OEE-beräkningen.



Figur 11. **Operatören kan övervaka produktionsdata**

6.1.3 Avslut av order

När produktionen för ordern är klar stoppar operatören den fysiska produktionen. Stoppet registrerar operatören som ”Ställtid/Ny order”. Det betyder att ordern är klar och produktionsdata för ordern loggas. När stoppet registreras som ”Ställtid/Ny order” byter klienten till nästa order. Den produktion som kommer efter att en order är klar registreras till nästa order och statistik korrigeras efter den orderns produktval.

Om stället går för snabbt för att systemet skall hinna reagera kan operatören manuellt skifta mellan olika order.

Om det är orderbrist eller skiftet skall avslutas väljer operatören ”orderbrist” som stopporsak. Det innebär ett planerat stopp som inte påvverkar OEE-värdet.

När skiftet är slut, avlutas det med knappen ”avsluta skift”. Om en order är pågående kommer ordern att kunna fortsätta produceras nästa skift. Eventuell produktion när ett skift inte är pågående kommer inte att registreras till någon order. För skiften finns det möjlighet att koppla information som kan användas för spårning av effektivitetsproblem. Skiftet kan exempelvis kopplas till en användare för att kunna undersöka hur olika operatörer påverkat effektiviteten på produktionsavsnittet, tillägg avsnitt 9.6.

6.2 Avrapportering

När ett produktionsavsnitt slutar producera aktiveras ett stopp. Rapporteringsklienten påkallar då operatörens uppmärksamhet, för att rapportera orsaken till problemet, se Figur 12. När ett stopp inträffar kan operatören för produktionsavsnittet först fokusera på problemet. Innan återuppstart eller kort efteråt registrerar operatören orsaken i rapporteringsklienten. Orsaken till problemet som väljs av operatören definierar hur stoppet behandlas i statistiken. Dels summeras de olika orsakerna ihop, men orsaken definierar också om stoppet är planerat eller oplanerat. För OEE-värdet är det skillnad på planerat och oplanerat stopp. Oplanerade stopp påverkar OEE-talet negativt. Planerade stopp dras från tillgänglig anläggningstid och ger planerad produktionstid, se kapitel 3.1.

Om ett stopp blir felregistrerat har operatören en möjlighet att omkategorisera de senaste stoppen. Det görs i fliken stopp.

Problem med att operatörer inte registrerar stopp eller tar lång tid innan de registrerar stopp hanteras av användargränssnittet. Systemet kan låsa användargränssnittet för att operatörerna skall tvingas rapportera stopp. Om orderna är stora så att ingen uppmärksamhet från operatören behövs under dagen kan det ändå uppstå problem med att operatörerna inte registrerar stoppen. Då kan andra åtgärder införas där den fysiska processen hindras från återuppstart innan stoppet är registrerat. Tillägg avsnitt 9.3. Ett alternativ är att låta stoppet registreras som övrigt om ingen har definierat någon orsak innan nästa stopp. Då är det lätt att följa upp och uppmärksamma problemet för operatörerna om många stopp definierats som övrigt.

Figur 12. Operatören rapporterar varför produktionen har stannat.

6.2.1 Rast

Raster är en stopporsak som ofta följs av ett annat stopp som till exempel underhåll eller ställ. Beroende på företagskultur och produktionslayout kan rasterna behandlas på olika sätt i systemet. Om rasterna är flexibla och det finns en variation i längden på rasterna skall rasterna bara kvitteras. Det görs genom att rasten registreras efter att operatören är tillbaka från rasten eller genom att produktionen startar. Om rasten har kvitterats manuellt kommer systemet att registrera att ett stopp fortfarande är aktivt. Det medför en möjlighet att registrera efterkommande tid som exempelvis ett underhåll eller justering.

Om rasterna är fasta och eventuell extra rast tagen av en operatör skall betraktas som ett oplanerat stopp finns möjlighet att tidsbestämma raster. Registreringen av stoppen fungerar på samma sätt som för flexibla raster. Tiden under rasten som överstiger den fasta tiden bedöms som övertid för rasten och kan behandlas som ett oplanerat stopp.

6.2.2 Manuell störning eller låst störning

Under ett underhåll eller en reparation kan det vara nödvändigt att låta systemet registrera produkter som helt kasseras eller inte existerar. Möjligheter finns då att spärra systemet från att registrera produkterna. Operatören får då aktivera ”manuell störning” eller ”låst störning”. Under ett pågående stopp kan operatören när som helst låsa stoppet (Låst Störning). Det innebär att stoppet är aktivt till dess att stoppet låses upp och produktionsavsnittet är aktiv. Under pågående produktion eller innan ett stopp hunnit bli registrerat kan manuellt stopp aktiveras. Manuellt stopp ignorerar på samma sätt som låst stopp all efterkommande produktion och låter stoppet vara aktivt till dess att stoppet låses upp av operatören.

Exempel på låst störning

Ett Stoppet inträffar, en avrapporteringsruta visas på skärmen och operatören avrapporterar störningen. Om operatören sedan vill att stoppet skall ignorera signaler klickar han på knappen lås störning. Systemet låter då stoppet vara aktivt tills både störningen har låsts upp och processen är igång. Ingen produktion eller några nya stopp registreras.

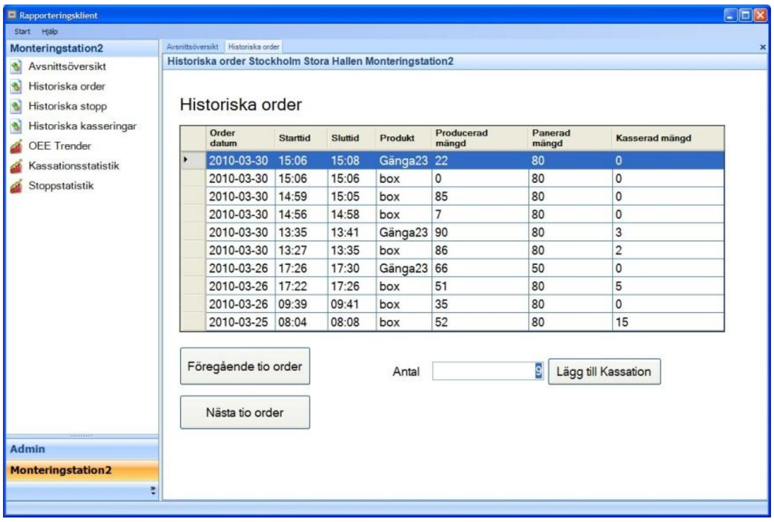
Exempel på manuell störning

Om ett underhåll skall påbörjas innan en avrapporteringsruta aktiveras trycker operatören direkt på knappen ”manuell störning”. Knappen startar en störning och slutar logga alla aktiviteter. Den manuella störningen måste också avrapporteras med orsak.

6.3 Kassation

Kasserad produktion anmäls av operatören eller registreras automatiskt. Manuell kassation registreras i ett fält på maskinöversiktssidan, se Figur 12. När operatören bekräftat en kassation syns avrapporteringsruta för kategorisering av kassationen. Den är byggd på samma princip som avrapporteringsruta för registrering av stopp. Kassationerna kan grupperas eller vara direkta.

Har ordern redan blivit klar kan operatören gå tillbaka i historiken och lägga till en kassation, se Figur 13. Om en kassation blir felregistrerad, har operatören en möjlighet att omkategorisera de senaste 10 kassationerna. Det görs i fliken kassation.

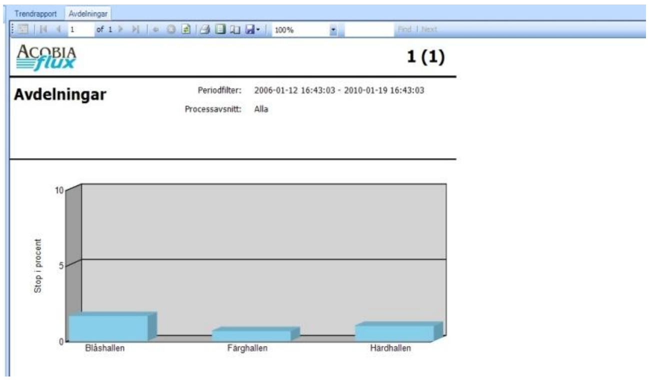


Figur 13. Kassationer kan även registreras i efterhand.

6.4 Statistik

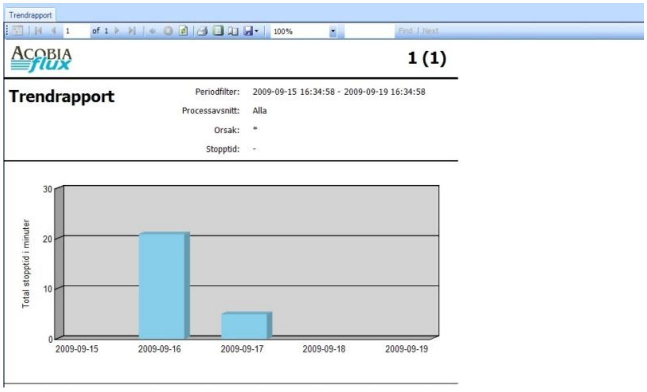
I rapporteringsklienten finns även tillgång till relevant statistik. Vilken statistik som finns tillgänglig kan bestämmas av kund och konfigureras olika. Exempel på statistik att visa i rapporteringsklienten är:

- OEE, Trender för en station, avdelning eller enheter.
- Dagens topplistor för produktionsavsnitt i företaget.
- Närliggande avdelningar i företaget jämförs för att skapa en intern tävling om att bli bästa avdelningen, se Figur 14.



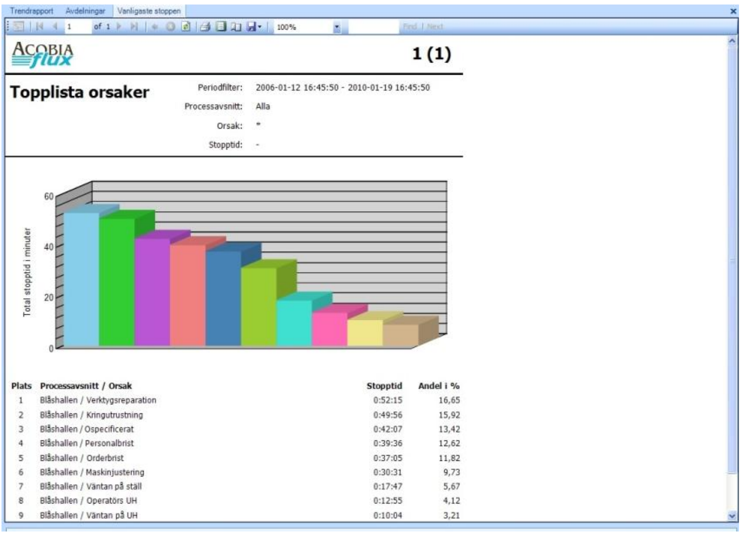
Figur 14. Avdelningarnas stopptider i relation mot produktionstid.

- Trendrapporter för stopp per dag ger operatören en återkoppling om hur stabil processen är, se Figur 15.



Figur 15. Antalet oplanerade stoppminuter per dag.

- Operatören kan bli mer vaksam på de vanliga problemen i produktionsavsnittet, om det finns en topplista över vilka stopporsaker som är vanligast, se Figur 16.

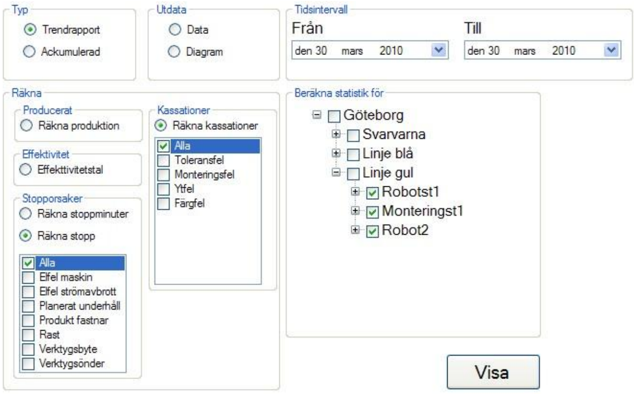


Figur 16. Vilka fel som orsaker att produktionen stannar mest.

7 Uppföljningsklient

Via ett webbgränssnitt får användarna statistik över produktionen. Återkopplingen gör att problem i produktionen är lättare att ta lärdom av. Produktutvecklingsavdelningen kan se vilka typer av produkter som oftast skapar problem i produktionen. Produktionsavdelningen får information om vilka produktionsavsnitt som har de största problemen. Dessutom är det lätt att göra uppföljningar och se om utförda åtgärder bidrar till minskade problem.

I uppföljningsklienten går det att filtrera och visa statistik för produktionsenheter, avdelningar, enstaka produktionsavsnitt, produkter, stopporsaker, kassationsorsaker och tidsperioder, se Figur 17. För de olika enheterna kan flera saker beräknas, effektivitetstal, antal stopp, stoppminuter, kassationer och producerad mängd. Dynamiken gör att rapporter kan skapas som visar trender i många dimensioner.



Figur 17. Användaren väljer vilken data som skall presenteras.

Det går även att visa valda delar av aktuell statistik i realtid med några timmars historik. Då uppdateras webbgränssnittet automatiskt så att realtidssiffror kan visas på exempelvis en tv-monitor i fabriken eller lunchrummet.

8 Kravsammanfattning

Systemet kan:

- Registrera och spara information om produktionsstopp.
- Registrera och spara information om produktkassationer.
- Registrera och spara information om produktionshistorik.
- Beräkna och presentera effektivitetstal, OEE-tal.
- Presentera statistik över den sparade informationen.
- Byggas ut med fler klienter utan att påverka befintlig data.
- Administreras från ett webbgränssnitt.
- Visa statistik från ett webbgränssnitt.
- Ingen eller liten påverkan på produktionen.

Systemet är inte lämpat för eller kan inte:

- Styra processen.
- Arbeta i en annan miljö än en Windows-miljö.
- Om rapporteringsklienten inte kan placeras i direkt anslutning till operatörens arbetsplats.

9 Tillägg

Eftersom systemet är konstruerat modulärt och med tanke på att det skall vara möjligt till utbyggnad beskrivs här ett par förslag med tillägg.

9.1 Kalenderfunktion

En implementering som gör det möjligt för administratören att planera tillgänglig anläggningstid för produktionen i förtid. Det innebär att planerad produktionstid kommer att få ett bättre underlag. Men det innebär också ett kontinuerligt administrativt arbete för företaget.

9.2 Manuell inmatning och justering av börvärdet på rapporteringsklienten

Funktion för att justera börvärdet per order i rapporteringsklienten. Inmatning sker genom att klicka på börvärdet och sedan använda en touch inmatningskontroll. Implikationer detta tillägg ger beskrivs i kapitel 3.2

9.3 Tvingad rapportering

I en process med stora order som innehåller flera stopp innan ordern är klar kan det bli problem att operatörer väntar för länge med att registrera stoppen. Åtgärder kan då införas där den fysiska processen hindras från återuppstart innan stoppet är registrerat. Det innebär att en styrsignal till processens styrsystem behövs. Tvingad rapportering kan användas som ett sätt att få korrekt registrerad data.

9.4 Automatiska fel

Om det tillgängliga systemet klarar att upptäcka orsaken till stoppen i produktionsavsnittet kan man låta systemet registrera stopp automatiskt. Det går att låta vissa stopp bli automatiskt loggade medan andra odefinierade stopp registreras som vanligt.

9.5 Småstoppsövervakning

Det är svårt att upptäcka små stopp i en automatisk produktion. Små stopp är lätt åtgärdade av operatören och uppmärksammas därför inte så lätt som ett stort eller långt stopp. Men samtidigt har de små stoppen en stor inverkan på hastigheten i produktionen. Det är därför viktigt att försöka hitta de små stoppen. För att kunna uppmärksamma små stopp med Efficiency Manager krävs att en extra signal från styrsystemet som talar om när produktionen står stilla. Stoppen kommer då kunna registreras som vanligt men om stoppet är snabbt åtgärdat blir det inte statiskfört under tillgänglighet. I stället blir stoppen effektivitetsförluster precis som om inte tillägget är med. Skillnaden är att de små stoppen kan följas upp och visualiseras.

9.6 Skiftregistrering

Det är möjligt att spara information som hör samman med olika skift. Exempelvis kan man undersöka hur mycket det påverkar effektiviteten beroende på vem som är operatör på produktionsavsnittet.

10 Referenser

Esping, M. (den 07 12 2009). Exempel - Efficiency Manager. Göteborg: AcobiaFLUX.

Jonsson, P., & Lesshammar, M. (1999). Evaluation and improvement of manufacturing. *International Journal of Operations & Production Management , 19* (1), 55-78.

Jönsson, A. (den 30 12 2009). Funktionsspecifikation Störningshantering. Göteborg: AcobiaFLUX.

Nakajima, S. (1988). *An Introduction to TPM.* Portland: Productivity Press.

Vorne Industries. (2008). *The Fast Guide to OEE.* Itasca, IL USA: Vorne Industries Inc.