INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

DESIGN OF AN INTEGRATED FRAMEWORK FOR DIAGNOSIS AND MAINTENANCE OF MANUFACTURING SYSTEMS

Alin-Gabriel Mihoc, Laurent Deshayes, Khalid Kouiss

Institut Français de Mécanique Avancée Campus de Clermont-Ferrand Les Cézeaux BP 265-63175 Aubière Cedex Email: laurent.deshayes@ifma.fr

ABSTRACT

Today's manufacturing equipment increases in complexity and size which makes more difficult the identification of potential problems and generating maintenance procedures. In addition, manufacturing companies use many different generations of equipments that makes more difficult their maintenance. All these equipments represent an important, growing capital investment that makes necessary to develop procedures to locate and solve problems. However, often solving problems require an expert engineer or technician with specific know-how and knowledge about the equipment. But such an expert is not always available or is simply busy with other tasks that make the manufacturing system unavailable for a costly time. After a bibliography review about maintenance systems and the management of their knowledge, the purpose of this paper is to present a design of an integrated framework for semi-automating the diagnosis of failure and for assisting the generation of maintenance procedures. This framework is based on the integration of three applications: a Computer Aided Maintenance (CAM) system, a Knowledge Base and a Supervisory Control And Data Acquisition system (SCADA). The knowledge base is the core of the framework and was built using an ontology developed under "Protégé". This knowledge base includes the use of default tree to automatically generate maintenance procedures, but also some maintenance control tools to update failure rates including micro-failures. Finally, a study case from a soap production plant is used to illustrate the framework.

Keywords: Maintenance, Diagnosis, Knowledge base, machining systems.

1 INTRODUCTION

The purpose of industrial maintenance is to insure that production systems are in working order and to maintain their functions throughout their life cycle. Lately, industrial maintenance is becoming one of the most important strategic functions in industry. It becomes a part of the continuous improvement policies. In addition, methods and practices of maintenance departments are in constant change due to the relentless technological development, the development of new management methods and the increasing need for reducing production costs.

For a long time the only purpose of maintenance departments was to repair machines once they failed. In today's businesses, the maintenance function must allow the prediction of anomalies in the production system and prevent any downtime. Because of such constraints, new integrated management tools at the execution and control level are required to optimize maintenance operations.

Integrated systems are required to optimize preventive maintenance which allows the prediction of failures and trigger maintenance operations at optimum moment. This is mainly due in part because many faults can still not be solved by predictions and reliability analysis based maintenance management. As a consequence, management methods such as the TPM (Total Production Maintenance) are used in order to allow the production department to perform low level maintenance operations. TPM reduces the number of micro-failures due to simple adjustments (that can be made by machine's operators) and in consequence may help to reduce and detect defaults before they occur. As an example, TPM may detect vibrations on equipment; by identifying such default the maintenance team is informed before a breakdown occurs. This permits to optimize both maintenance costs and production stops. As a consequence, in such policies, the maintenance department is able to better

focus its expertise and time in order to optimally maintain the manufacturing system by respecting quality and production constraints.
As a consequence of all these major changes, integration of maintenance tools into the Manufacturing Execution System (MES) becomes an increasing need for optimizing industrial maintenance. MES is a recent concept whose objective is to build an information system for shops and to bridge the gap between planning tools such as ERP (Enterprise Resource Planning) and control systems. In fact, an MES is a collection of hardware/software components that enables the management and optimisation of production activities from order launch to finished goods. While maintaining current and accurate data, an MES, initiates, responds to and reports on plant activities as they occur [1]. We can include in the scope of MES several functions that support scheduling, supervision, maintenance and execution activities. The maintenance function has been defined as one of the eleven functions of the MES [2]. Maintenance is reaching such importance that it implies not only to repair the system and maintain it to its specified state, but also to improve the system and to include quality policies.
The objective of this paper is to propose an open architecture which will offer support for the optimization of preventive, predictive and corrective maintenance. The MES tools integrated by this architecture are SCADA (Supervisory Control and Data Acquisition) systems and CAMM (Computer Aided Maintenance Management) tools. This open architecture enables future developments for integrating additional MES functions. The architecture must allow, beside the detection and diagnosis of failures, the detection and analysis of micro-failures. A knowledge base was defined and it represents the core of the system. Finally a soap production plant is used to validate the framework.

2 MAINTENANCE’S ROLE IN MODERN PRODUCTION SYSTEMS.

Today’s consumer market is changing; product life cycle is shortening reducing strongly the delay to release a new product. As a direct consequence, workshops need to accommodate their manufacturing systems to manage an increasing number of references and decrease delays. In order to adapt and to survive in this changing environment companies need to optimize their workshops by better controlling performances and quality of the execution systems. Industrial maintenance plays a major part since its role is to maintain the execution system at constant specification. In many places, the role of industrial maintenance is changing from curative maintenance to preventive maintenance which makes necessary the use of computer aided tools like Computer-Aided Maintenance Management (CAMM) software. Maintenance becomes also a part of quality and improvement policies. In fact, it not only maintains the system to its optimal conditions, but it also participates in improving the system and quality procedures.
Of course, a direct consequence of such a role for industrial maintenance is the need to integrate information provided by all stages of the life cycle of the system, from equipment designers, to machine removal. During the execution phase, the industrial maintenance applications exchange, through the Manufacturing Execution Systems (MES), information with the different operative functions. Such communication is the prerequisite for constantly improve and optimize the entire production system.
But as shown by many applications, management methods are needed to exchange information between entities. For maintenance, this exchange was partially reinforced by the TPM (Total Productive Maintenance) policy since the 80’s by proposing series of steps to ensure that production and maintenance departments work together harmoniously. Many of TPM tools are manual instructions and spread sheets. There are few full integrated maintenance based systems within the MES**.**
Another important aspect of maintenance is the high expertise level needed for solving failures and reducing their frequency. Indeed, at this level very specific languages are used by operators and maintenance technicians that make difficult to install an automatic treatment of information. Therefore, semantic representations of procedures and diagnosis methods are necessary to unambiguously exchange information between heterogeneous systems with different specific languages. For instance, the term “failure” has different meanings given by either standards, publications, and even across the company. For Isermann and Ballé [3] a “failure” is: ”a permanent interruption of a system’s ability to perform a required function under specified operating conditions”. For the NF EN 13306 standard [4] a failure is a “deterioration or suspension of the aptitude of a system to be achieved its necessary functions with the performances defined in the technical specifications”. For this definition a degradation of the system function, even if the production is still

working at its optimal value, is considered as a failure, which makes more difficult the measure of failures downtime. This is quite different Isermann and Ballé definition for which a failure is strictly when a system's function is not anymore active. This definition is much easier to measure failure duration. Such differences require formal representation of their semantic into the knowledge base.
Considering Isermann and Ballé definition [3], failures are generally well taken into account because their effects are long lasting and cause machine stops. On the other side, micro-failures are not so well accounted due to their short lifespan (often under a minute). But micro-failures may have a significant effect on production rates, more particularly, when many product formats must be produced on a same industrial manufacturing system. In fact, to automatically estimate micro-failures duration information from SCADA system must be coupled with some knowledge rules in order to identify possible causes, to choose the right maintenance procedure and to smartly update all ratios necessary for maintenance and production management. This is more detailed in the next section.
In all cases, once a failure or a micro-failure has been detected, well defined procedures are needed to reestablish the failed function. The procedure can be automated or manual and is executed by an expert technician. A maintenance procedure includes two distinct parts: diagnostics and operative procedure. Diagnostics are not new to the industry, many attempts to integrate them in expert systems were made since the 80's. One particular and interesting method is presented in [4] by using a tree structure similar to the failure tree to diagnose mechanical systems. This method requires a good knowledge of the system's components and the probability of each node to fail must be updated each time the component fails. Supervisory control tools are very useful to update failure data. But, procedures are dynamic along the time of service of the system. They may be modified in order to take into account improvements on the manufacturing systems, to optimize operative procedures… Therefore there is a strong need to dynamically generate procedures adapted for each situation. This is rarely discussed in bibliography.
Keeping in mind all the elements presented so far and extrapolating from several state of the art papers [5, 6, 7], we concluded that an integrated maintenance system must be designed considering the following characteristics and roles:

- collect the knowledge and make it available to the MES functions and therefore to the entire enterprise,
- combine expertise from different fields,
- distribute the expertise inside the MES to the concerned company services,
- working in a collaborative environment, thus allowing expertise from different domains to interact,
- training less qualified technicians to improve their knowledge,
- integrate essential quality policies,
- add the machine operators in the role of industrial maintenance,
- be ergonomic.

However, and in order to maintain the expert system, the human component plays a crucial role and intervenes at many stages, from updating knowledge database to performing the required tasks. Therefore, it is included in the integrated architecture that is presented next.

3 INTEGRATED MAINTENANCE SYSTEM'S ARCHITECTURE

The general view of the system's architecture is presented in Figure 1. SCADA's role is to supervise the production system and inform the user of changes, failures and the occurrence of predefined alarms. It has limited decision capabilities and focuses on the execution systems. CAMM stores in its database the characteristics of every component found in the production systems as well as reliability data such as FMECA (Failure Modes, Effect and Criticality Analysis) and failure trees. Procedures, for components having a preventive plan, are triggered thanks to counters. CAMM generally presents few diagnosis capabilities. It stores both numerical formats of text and images procedures to follow to solve a fault on a system.
A knowledge base is used to store and deduce rules and knowledge based on maintenance procedures analysis (generally issued from CAMM systems), SCADA and operator knowledge about failures including the tools necessary for TPM and quality control. The connections made from the SCADA (to CAMM and Knowledge base) are unidirectional: the supervisor sends at predetermined intervals the counters towards CAMM and each time a failure or a micro-failure occurs it notifies the knowledge base. The knowledge base requests procedures and part lists from the CAMM database, it

creates work orders, when necessary updates reliability data. The class diagram presented in Figure 2 presents all functions that may accomplish each system and the information they communicate to each other.

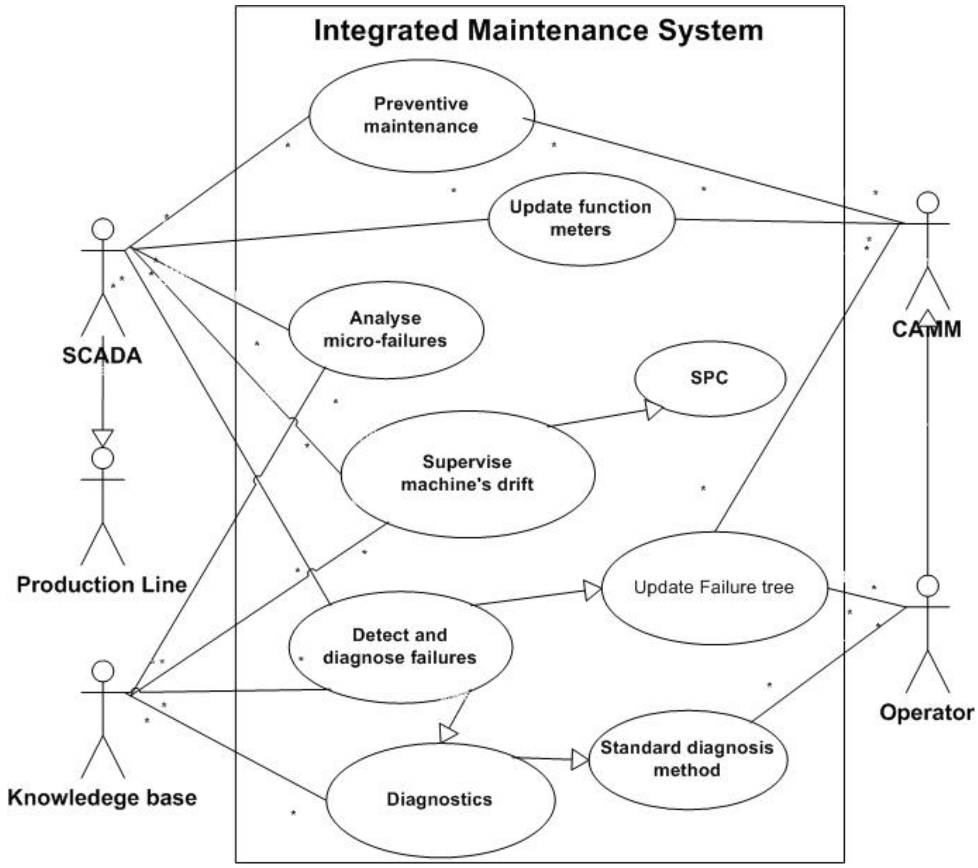

Figure 1: Use case of the integrated maintenance architecture

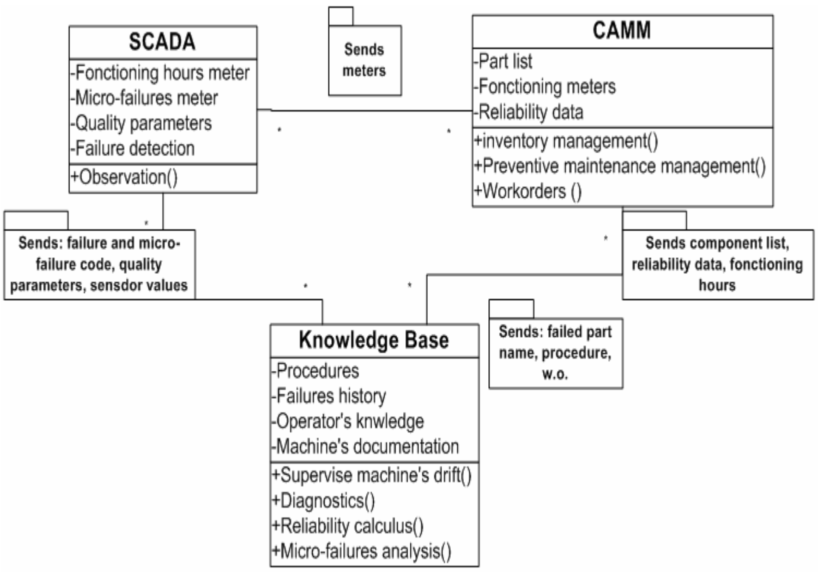

Figure 2: Class diagram of the architecture

3.1 Micro-failures

From the point of view of the SCADA system, micro-failures can be placed in two categories: 1) detectable, and 2) non-detectable. It must be stated that for the SCADA application there is no difference between a failure and a detectable micro-failure, they are treated in the same way. As soon as they are detected the knowledge base is notified and data concerning its duration and number of occurrences so far are recorded and transmitted.

Detectable micro-failures are either directly detectable because there is a sensor dedicated or determined by combining values from different sensors. SCADA applications were not designed to detect micro-failures; however, they were designed to detect failures. So in order to get data about micro-failures they are defined as failures. This is the role of the knowledge base to identify the micro-failure and its duration based on expert's rules.
Figure 3 presents a scenario for the treatment of micro-failures. When a detectable micro-failure occurs on the production system, SCADA increments its counter for that particular micro-failure and records its duration. Once the alarm is acquitted, the information collected previously is sent to the knowledge base for determining the micro-failure's frequency. If the frequency is above a certain limit value then the machine needs to be retuned otherwise nothing is done. If there is a need for an intervention, then a work order is generated for the maintenance operator trough the CAMM system.
Non-detectable micro-failures cannot be detected by the SCADA, so the system must rely on information from the operator. TPM tools are very useful to manually count the frequency of each micro-failure. By combining average micro-failure duration, the total unproductive time due to the micro-failure may be deduced. The frequency is also used to determine the necessary corrective actions on the equipment.

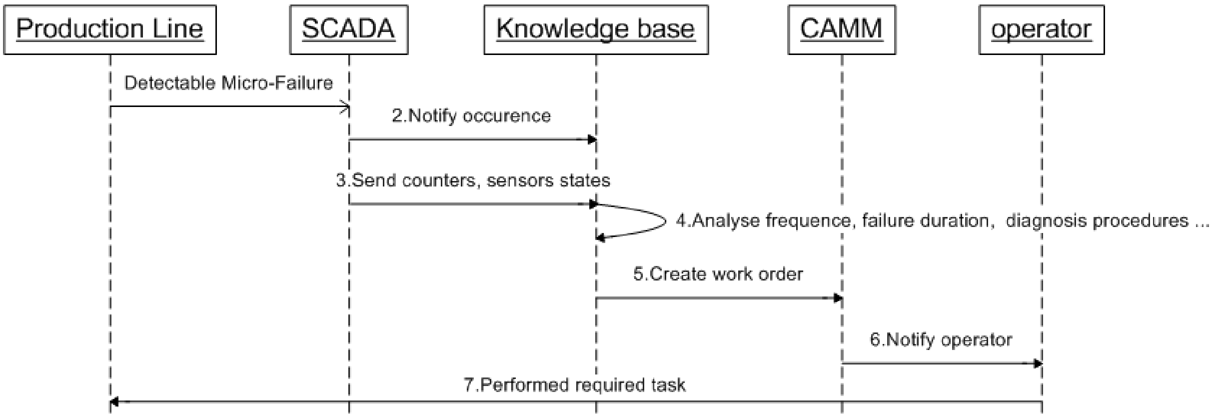

Figure 3: Sequence diagram for the treatment of detectable micro-failures

The actions to treat non detectable micro-failures are similar to the ones previously presented in Figure 2. Instead of a SCADA system, an operator is used to determine the fault and to take the decision. The knowledge base communicates the number of occurrences when an intervention is needed; in order to do so, data on occurrences must be introduced regularly.

3.2 Diagnosis

Diagnosis is more and more used to determine the cause of failures. Its importance depends mainly upon the machine position in its life cycle. A classical case is to consider that at the beginning and the end of the equipment life cycle, many failures are new to the operators so a diagnosis tool is needed to help them. During the maturity of the machine, failures are better known and may be better predicted. As a consequence diagnosis is needed occasionally during this period and new failures are rare. But when unexpected failures occur, they are often extremely costly.

For the failures already treated in the past, "if … then" rules are used and combined to failure trees to determine the cause of the failure and their associated maintenance procedure. Failure trees are often used to make a list of the components most probable to have failed. The advantage of using failure trees is that it is generally easily found in the machine's documentation or can be built based on technical documentation. In order to determine the probability of a component to fail, each node of the tree has a probability of occurrence which is the component failure rate λ given by the following equations:

$$\lambda = \frac{1}{MTBF}(\,failures / hour) \tag{1}$$

$$MTBF = \int_0^\infty R(t)dt(hours) \tag{2}$$

where the MTBF (Mean Time Before Failures) is calculated using the reliability function R(t)in (2). R(t) is estimated using the machine's failure history. By assuming that the reliability is a distribution function of a random variable, the probability that a certain component has failed after T_f hours is given by.

$$\lambda = \frac{1}{\int_0^{T_f} R(T)dt} \tag{3}$$

The diagnostics program uses the failure tree to make a list with the components that have the highest probability to have failed, above 50%. This list is then feed to the theoretical model part, if the component has one, or presented to the operator. As presented in Figure 4 the theoretical model program uses the sensor readings to compare the theoretical response of the component to the same input as the real component is subjected, if the experimental response is not in a certain interval of the theoretical response, then the component is faulty and must be replaced. If after this step no cause for the failure is found, then a standard diagnosis method is initiated which will guide the technician to determine the failed component(s). Once the repairs are made, the operator creates an intervention report which contains the procedure he used or created to eliminate the failure. Then from this report an "if … then" rule is created and stored in the knowledge base for future usage.

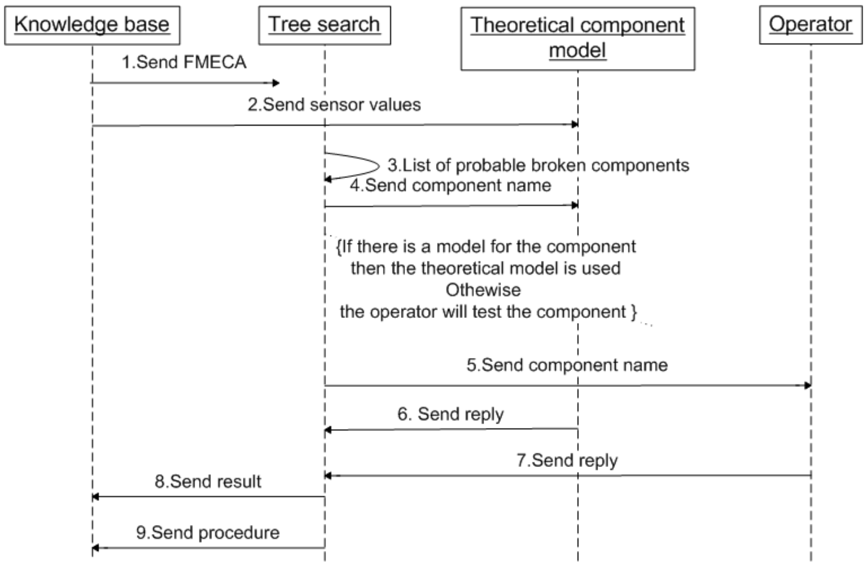

Figure 4: Sequence diagram of diagnosis analysis on components

3.3 Reliability function update

The diagnosis function uses reliability data in order to make a correct diagnosis so it is vital to keep the information up to date. This is done either by hand or using CAMM systems. Updating the reliability data is made automatically each time the diagnosis function is used or a failure occurs on the production system. Figure 5 shows the sequence of operations: once a failure occurs on the production system the diagnosis program is run and once the faulty component is identified the diagnosis program adds the new data to the machine's history and recalculates the reliability function.

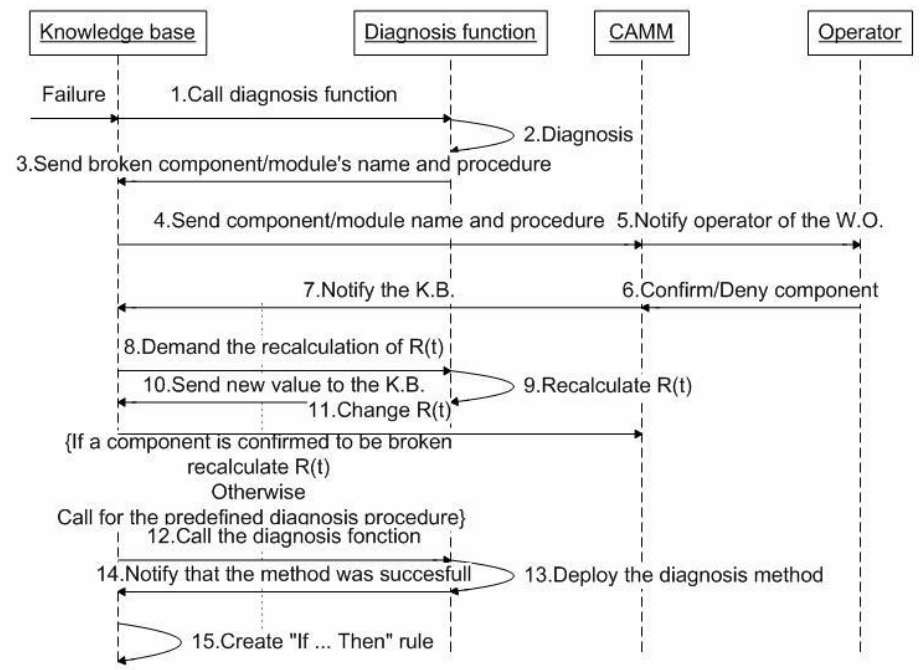

Figure 5: Sequence diagram for reliability data update

3.4 Learning tool

All the procedures, failure types and their diagnosis are stored in the knowledge base. Some procedures are well known by the experienced technician; their storage and update ensure the reuse of the corporate knowledge by other operators. In Figure 6, two examples present the use of the system as a learning tool: one for displaying past data, i.e. the equipment history, the second one to display maintenance procedures.

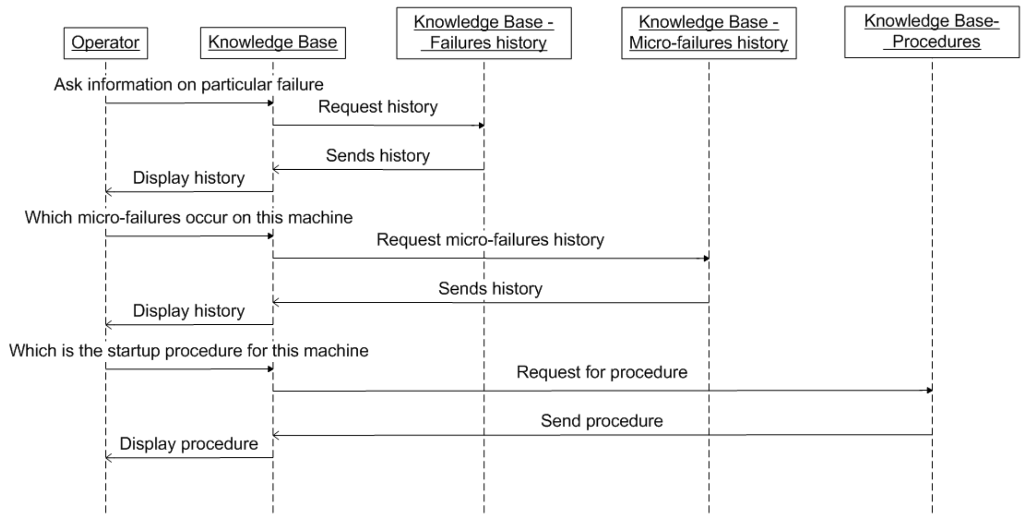

Figure 6: Sequence diagram for using the system as a learning tool

3.5 Supervising machine's drift

Nowadays, most workshops use Synthetic Efficiency Rates (SER) to picture workshop's performance in terms of quality, availability and machine performance. As discussed previously SCADA Systems present very interesting functions for maintenance but also to more precisely evaluate SER. Many imprecision on SER comes from the lack of precise measures of micro-failures times. In order to correctly calculate failure times, SCADA system need the use of rules to effectively select the right micro-failure type and then produce the adequate process parameter or maintenance procedures to apply. Figure 7 illustrates such scenario. At regular intervals SCADA sends to the knowledge base files containing the quality parameters logged since the last update. The knowledge base analyses the new data and provides actions to perform on the system. In the case where actions on the system are required, the information is forwarded to the CAMM to generate a Work Order (W.O.). The operator executes the required tasks and the knowledge base is updated.

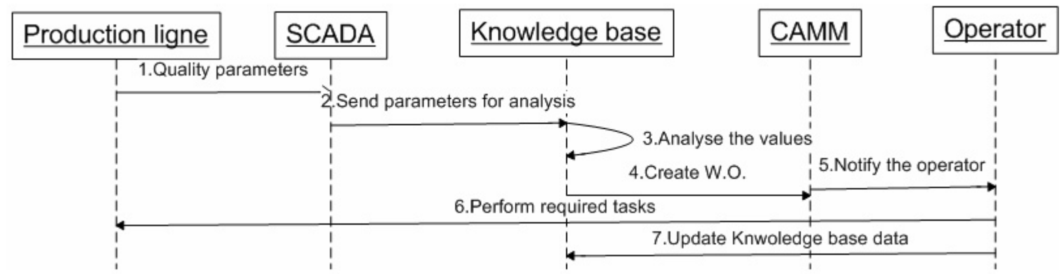

Figure 7: Sequence diagram of supervisory role

4 KNOWLEDGE BASE ARCHITECTURE

As stated in the previous sections, the proposed integrated maintenance system must respond to every change on the equipments of the production system. An ontology, created using the method presented in [8], is used in the knowledge base. Since most components are common for all types of industries the ontology is useful for multitude of applications and it can be tailored to the particular needs of a workshop.

The ontology used in the knowledge base must reflect the equipment structure and their functions in the production system. The taxonomic structure presented in Figure 8 is used. At the top level the concept "machine" is the most abstract term used in the taxonomy. On the next level "component(s)" describing the sub-equipments of the machine are them-selves divided into subcategories corresponding to the technological components such as electrical, hydraulic, pneumatic, and mechanical equipments. On the same level with "component" the concepts of failure, micro-failure and operator are used. They obviously are not machines but the notions and the properties defined for them are needed for the functions mentioned in the previous section. A procedure class contains all the procedures not related to a failure (startup and shutdown procedures for instance). Procedures for failures are pointed into the CAMM system avoiding a double copy of the information. They are also used by the expert system to automatically update the knowledge base. The next concept, which is also on the same level with "component", is the "auxiliary" class for consumables like filters, cooling liquid, oil, and so on, for which there is a programmed maintenance and no reliability variable are related to them. This knowledge is still required and could have been placed in one of the categories mentioned before but in order to simplify data processing, they are kept separately.

Concerning failures, the concept "failure" is associated to symptoms, faulty component(s) and procedure. "Micro-failures" have a frequency and type (detectable and non-detectable). The "operator" has a name and he is authorized to perform certain types of interventions. "Procedure" has a title and the last time it was updated and "auxiliary" components have a CAMM reference number and a name.

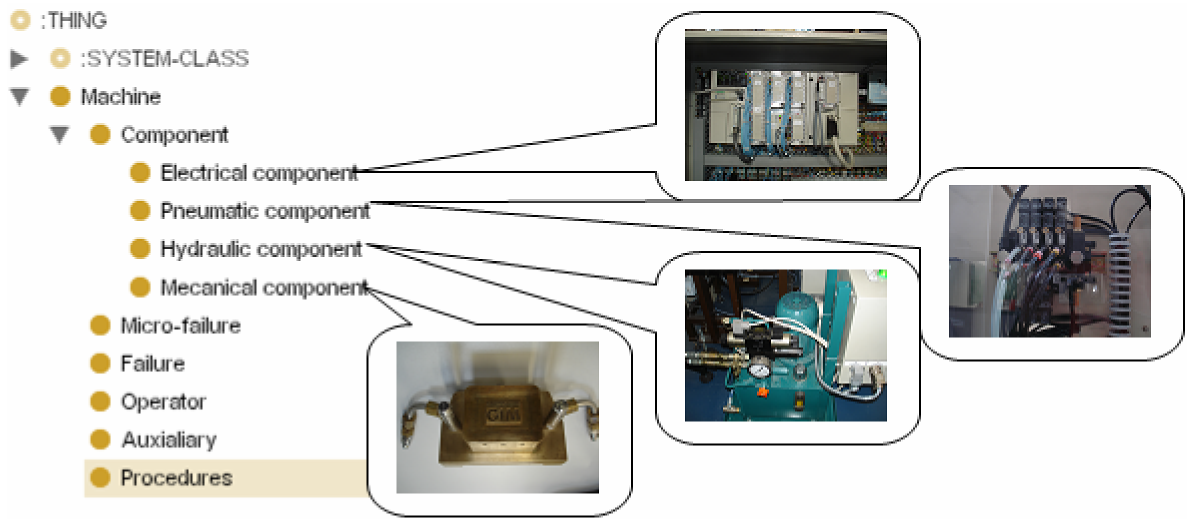

Figure 8: Taxonomy and examples of instances

5 STUDY CASE

5.1 General view of the SAVONICC production line

The architecture was partly deployed on the SAVONICC soap production line from the French PREMI (Regional Pole of Industrial Maintenance). The SAVONICC soap production line is an actual, functional production line presented in Figure 9, the main advantage of this production line is that it does not have the constraints of such a line under industrial exploitation conditions, it can be stopped any time to test and validate new procedures before being deployed to industry. The CAMM application used is MiniMaint and the SCADA application connected to the production line is PcVue.

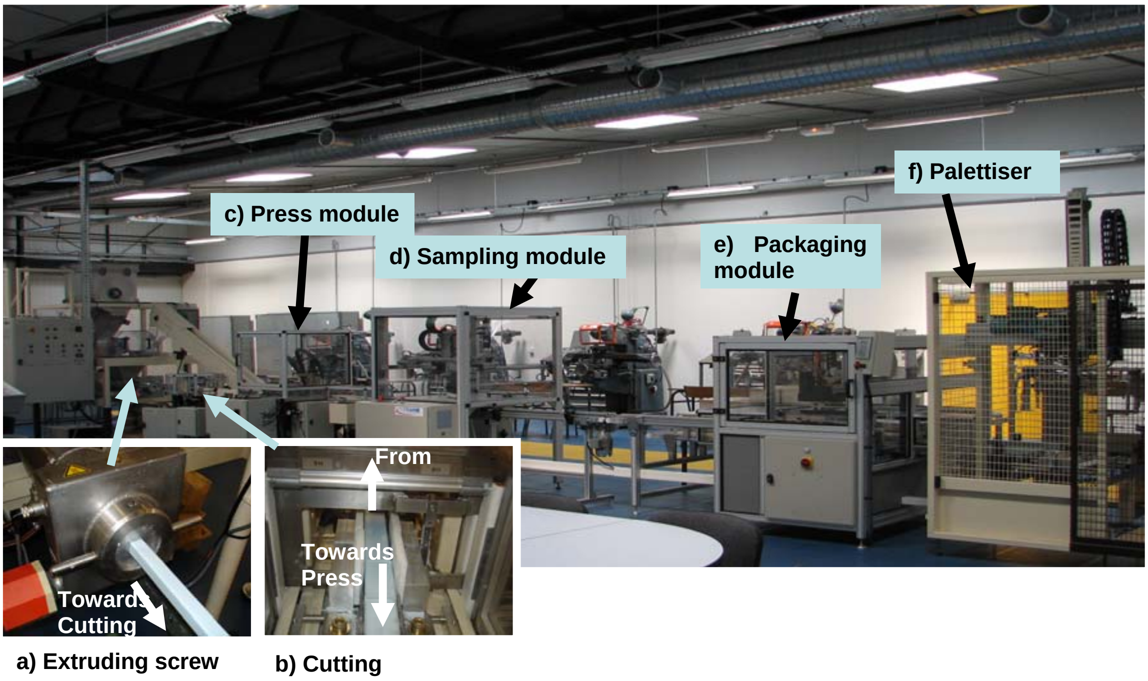

Figure 9: SAVONICC production line

SAVONICC is organized in five modules. The soap begins its transformation as soap flakes which are mixed with water and perfume and then it comes out as a continuous rectangular bar through the screw extruder Figure 9a), the cutting module (Figure 9 b) cuts the bar to block at the required length. The press module Figure 9c) molds the soap bars into the needed form and then sends the soap to weighing station. Finally, the packaging and palletiser modules Figure 9e) and f) finish the product by placing it in boxes and the boxes onto pallets.

5.2 Detailed study of the press module

For practical reasons the study was restricted to the press module, but can easily be generalized to all the production line. The press equipment is detailed in Figure 10a. The normal operation cycle for the press is the following:

1) the soap blocks are brought to the feeding actuator by the conveyor belt,
2) the presence sensor notifies the logic controller that a block is ready to be pressed,
3) the feeding actuator positions the block under the press
4) the press molds the block into the correct soap shape
5) the ejection actuator pushes the soap out of the press toward the weighing module and the cycle begins again.

Finally the system is supervised through three additional sensors: one at the press input indicated if there is a queue of soap blocks, two sensors placed next to the press to detect if a soap has adhered to one of the matrixes (superior or inferior). After analysis, it has been determined that the press module has three micro-failures classified in the two categories mentioned before.

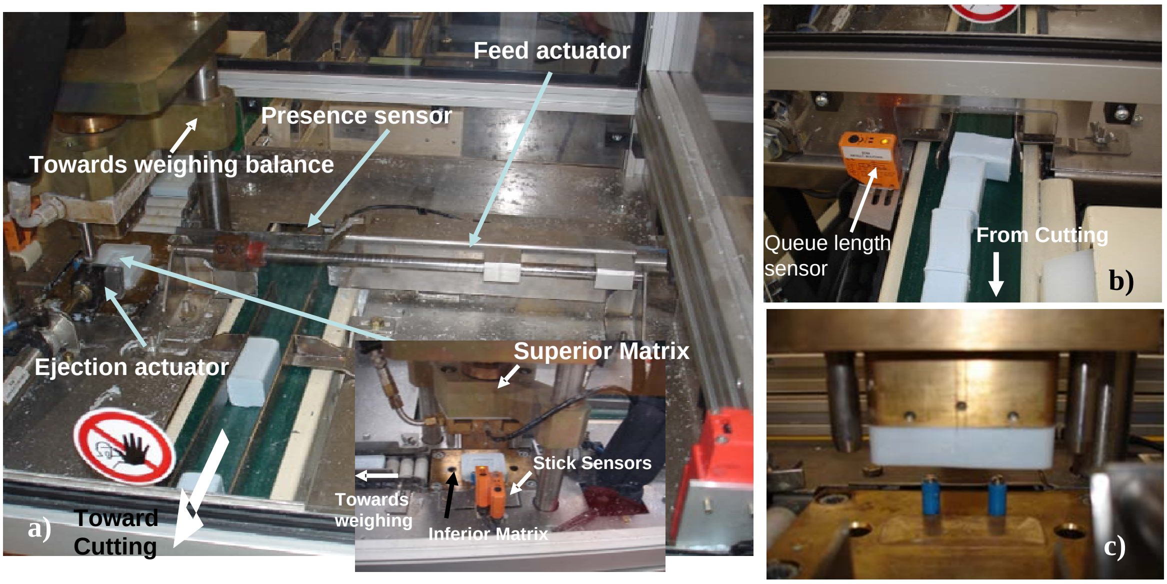

Figure 10: a) Press; b) Soap blocks the press input; c) Soap sticks to the upper matrix

5.2.1 Detectable micro-failure:

Two micro-failures are observed and detectable by the press sensors and SCADA. They are:

- soap sticking to either the lower or upper halves of the matrix, due to moisture or insufficient cooling, there are two sensors dedicated to the detection of this micro-failure, See Figure 10 c.
- soap blocks the entry to the press, see Figure 10 b. This micro-failure is not directly detected, it uses the combination of two sensors queue length and presence before feeding actuator. The detection is based on a knowledge rule using the information of sensors state. For the case of our press module, the following rule presented in Figure 11 is used.

```
 'The algorithm starts when the queue length sensor changes its status from 0 to 1
time.Date=SYSTEM("GETDATE"); 'get system date
       time.start=SYSTEM ("GETTIME"); ' system time, used for determining micro-failure
duration
While (queue length=1) do :
                if(presence sensor =0)
                Then
                         k=k+1;
                         If (k<3)
                                  then    time.stop=SYSTEM ("GETTIME");
                                           notify operator;
                                           write data to file;
                                           send file to knowledge base;
                                           Stop algorithm;
                                  else     DELAY (0.3);
                Else     k=0;
```

Figure 11: knowledge rule for determining duration of micro failure

5.2.2 Non-detectable micro-failures:

Figure 12a shows the effect of a soap sticking to the ejection actuator after the ejection actuator completed its action. On the return run the soap will detach from the actuator and will be crushed by the press in the next cycle causing either a false “soap stick to matrix” micro-failure detection or surface defects on the soap. Consequences are strong for product quality.

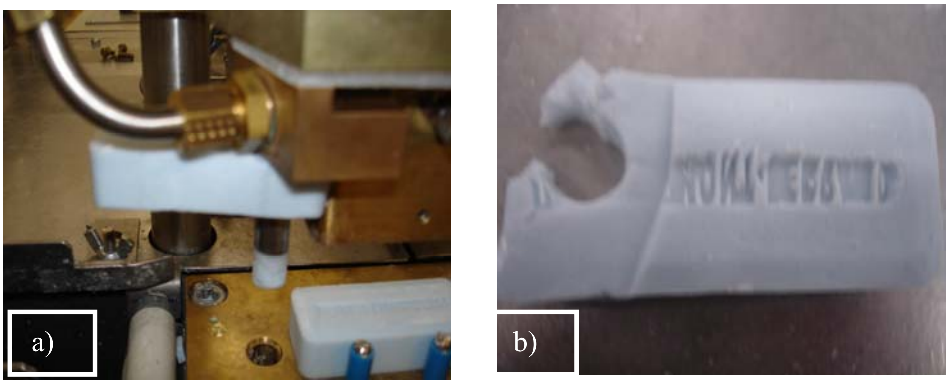

Figure 12: Soap crushed by the press

5.3 Knowledge base

The knowledge base was developed under Protégé [9], which is an ontology development tool that supports the creation, visualization, and manipulation of ontologies in various representation formats. It was developed in java and it is also open source so it can be customized to specific needs with ease. This ontology is directly part of the knowledge database.
The knowledge base has stored all the types of equipment the press module has and all the data related to micro-failures based on the ontology presented in section 4. Information about failures can be sorted by module, by subassembly or by part. Since other machines use similar parts that have the same generic name (i.e. pneumatic actuators) if the search is made using the generic name the result is a list with all the failures for that particular type of component which also show the measures taken on other machines thus helping him to make a decision faster and eliminating the trial and error method and reducing the need to use diagnosis methods.
For the detectable micro-failures functions were created for the user interface in PcVue. Using the variables defined in SCADA each micro-failure is detected, the occurrence counter is incremented and the duration is retained. The data is then sent to the knowledge base for analysis, at this stage using the micro-failure history the frequency is determined as well as the mean duration time. Using these values a decision is made either to do a maintenance operation on the machine or to wait until the frequency reaches a level when such an intervention is required.
Since for PcVue there is no difference between a failure and a micro-failure so the same function can be used to inform the knowledge base of the occurrence of new failures and send sensor values. The knowledge base makes the difference between a micro-failure and a failure using the data received and starts the appropriate function. Figure 13 shows a query made on the database to select micro failures occurring on the press module.

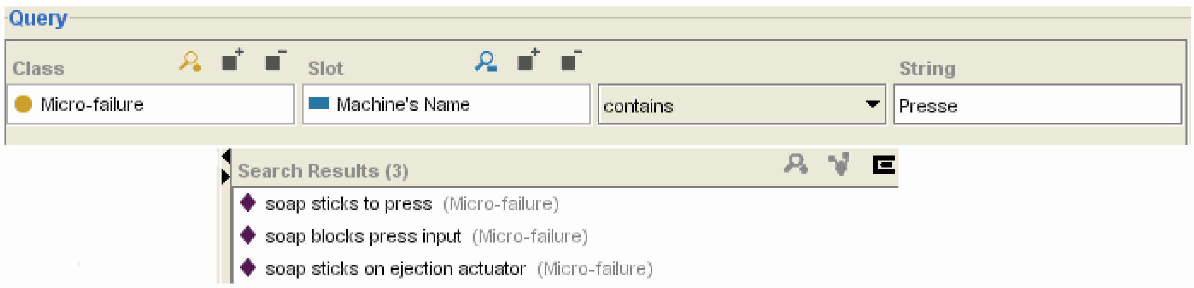

Figure 13: Query for all micro-failures by module and search results

5.4 CAMM application

The CAMM application updates its functioning hour's counters using an ASCII file. It must have a specific layout so a function was added to SCADA, which collects the necessary data, formats it, and writes the file on the local disk of the machine running the SCADA application. The CAMM is instructed to read the file on the knowledge base server.
CAMM systems generally store Text format of procedures that are not easily treatable by automated computer systems. With the evolution of the machines, the procedures must evolve as well and their

content enriched in order to treat new failures. Many procedures have parts in common, like the assembly and disassembly instructions, which are generally copied from older procedures and rarely updated. In some cases, for example when a part is replaced on many company equipments, the time needed to update all the procedures is consequent. Automating procedures updating and creation would help optimizing preventive maintenance operations and diagnosis. The next step of such an approach will be to use semantic analysis, based on the ontology we presented, and the knowledge base to automatically generate and update procedures adapted to each failure autonomously using rules like "**if** component A needs to be replaced **then** follow procedure B …".

6 CONCLUSIONS AND PERSPECTIVES

The integrated framework presented in this paper is designed to address all kind of maintenance procedures and policies of industrial production systems. It is designer to be integrated with all MES applications through a knowledge database that is the core of the architecture. In addition, this knowledge database ensures the integration of different MES systems using standard protocols.

In this study a SCADA and a CAMM system were implemented since they are directly connected to maintenance. An initial implementation for a soap plant validated the architecture and shows promising results.

As perspectives, the system needs more implementation to extend its functionalities to additional MES systems (such as MRPs, ERPs …) , but also to be able to use different kind of expert systems in order: 1) to improve automatic maintenance procedures editing using fault trees, and, 2) to ensure reliability based analysis for optimizing predictive maintenance. Theses aspects will be addressed in future work.

REFERENCES

[1] E. Wallace "*NIST Response to MES Request for Information*" NIST response to RFI-3, pp 1-24, 1999

[2] MESA International "*Controls Definition & MES to Control Data Flow Possibilities*" White Paper n°3, pp 1-7, 2000.

[3] Isermann R., Balle P., 1997,Trends in the application of model-based fault detection and diagnosis of technical processes, Control Engineering Practice,5/5:709-719

[4] NF EN 13306, Maintenance terminology, pp 20, 2001

[5] Guan J.,Graham ,1997,An Integrated Approach for Fault Diagnosis with Learning, Computers in Industry,32: 33-51

[6] Patel S.A.,Kammrani A.K.,1996,Intelligent Decision Support System for Diagnosis and Maintenance of Automated Systems, Computers ind. Engng, 30/2:297-319

[7] Liao S.-H.,2005, Expert system methodologies and applications – a decade review from 1995 to 2004, Expert Systems with Applications,28:93-103

[8] Noy N.F.,McGuinness D.L., 2001, Ontology development 101: A guide to creating your first ontology, available http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf (February 2007)

[9] Protégé available at http://protege.stanford.edu/ (February 2007)