



**HAL**  
open science

## On the use of SysML for Manufacturing Execution System design

Laurent Pietrac, Arnaud Lelevé, Sebastien Henry

► **To cite this version:**

Laurent Pietrac, Arnaud Lelevé, Sebastien Henry. On the use of SysML for Manufacturing Execution System design. ETFA 2011, Sep 2011, Toulouse, France. pp.1-8, 10.1109/ETFA.2011.6058984 . hal-00767991

**HAL Id: hal-00767991**

**<https://hal.science/hal-00767991>**

Submitted on 16 Jan 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On the use of SysML for Manufacturing Execution System design

Laurent Piétrac, Arnaud Lelevé  
Laboratoire Ampère (CNRS UMR5005)  
Université de Lyon, INSA-Lyon  
F-69621 Villeurbanne, FRANCE  
firstname.lastname@insa-lyon.fr

Sébastien Henry  
Laboratoire DISP  
Université de Lyon, UCBL  
F-69621 Villeurbanne, FRANCE  
sebastien.henry@univ-lyon1.fr

## Abstract

*In this paper, we show that the integration of a new MES within an existing manufacturing system requires to appeal to a method enabling the modeling of interactions between hardware and software components, the modeling of their behavior and their use by production staff. We show how SysML is a language fitting this need and we illustrate this discussion on a real sample.*

## 1. Introduction

Manufacturing Execution Systems (MES) are systems dedicated to manage the production execution. Located between the manufacturing system and company information systems (ERP, PLM, SCM). They are often introduced as an information system. Previous works on the MES have primarily focused on the integration of these different data models, such as the ISA95 standard which provides a UML class diagram for exchanges between a MES and an ERP [8]. Other scientific topics deal with architectures, such as agent based systems [4] or holonic approaches [3].

However, the MES must also be paired up with a manufacturing system. The constraints of this integration are different from those with the ERP. Indeed, data exchanged between the MES and other information systems are rich in information (work order, machine capability, ...) but the frequency of these exchanges is low (most often weekly). Exchanges with the manufacturing system are often made of binary information (sensor status, stock level, state of a given function), at most a few bytes, but their frequency is high as they occur on each state change [9].

Data provided by the manufacturing system to the MES affect directly its behavior. Similarly, the MES has also an impact on the behavior of the manufacturing system, for example via the production parameters that it transmits. A data model is not sufficient to guarantee the integration of a MES within a manufacturing system. The relationship between state changes of both MES and manufacturing system must also be considered, particularly through the chronology of exchanged data. Furthermore,

many changes are caused by events generated by the handled products and the actions performed by electrical or pneumatic actuators. To design the software part of the MES, every element of the manufacturing system, software (control programs) and hardware (products, actuators, sensors), must be known and modeled with their relationships.

Thus, the integration of a MES within a manufacturing system requires a modeling language able to represent software (MES and manufacturing system), hardware (manufacturing system) and their interactions. Today, the language best suiting this purpose is known to be SysML (Systems Modeling Language). It is a recent language derived from UML dedicated for systems engineering applications. These systems may include hardware, software, information, processes, personnel, and facilities. In this paper, we introduce, at first, MES concept and especially its activity of production performance analysis. After introducing SysML, a MES design approach based on this language is proposed. Our approach is then applied to an example before concluding on this proposal.

## 2. Manufacturing Execution Systems

### 2.1. MES position

Competition between manufacturers has always pushed them to maximize the performance of their production system. In industrialized countries and in mass production, the research of a better quality, of a reduced tick time, and of a lower unit cost has led to automate the process. On the other hand, the rationalization and the homogenization of commercial activities – customer relationship management, order management, stock management, etc – have led to replace specific softwares with Enterprise Resource Planning (ERP).

Between the ERP, which gives a vision of the company's activities at medium and long terms, and manufacturing systems which produce in real-time, the roles of the MES are to manage the execution of production operations necessary to achieve production orders, and to inform the ERP about these achievements and the real capabilities of the production system.

## 2.2. MES activities

In ISA-95 standard [7], the International Society of Automation defines a generic model of the eight activities of Manufacturing Execution Systems (see fig. 1):

- **Product definition management** manages all information about the product required for manufacturing;
- **Production resource management** manages the information about resources required by production operations;
- **Detailed production scheduling** includes local planning and scheduling of production and resources;
- **Production dispatching** manages the flow of production by dispatching production to equipment and personnel;
- **Production execution** directs the performance of work, as specified by the contents of the dispatch list elements;
- **Production data collection** collects and manages process and equipment information;
- **Production tracking** prepares the production response for the ERP;
- **Production performance analysis** provides feedback about production.

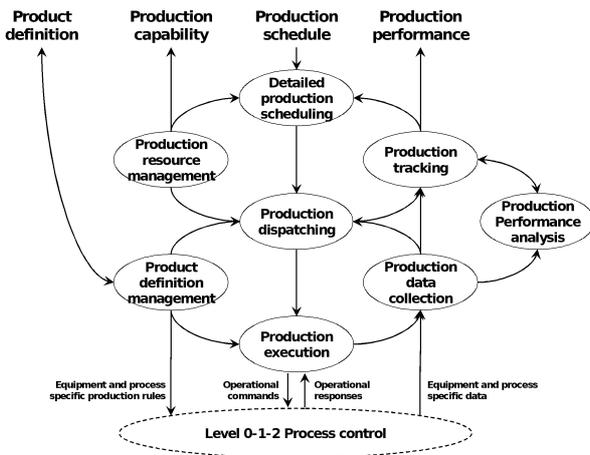


Figure 1. Activity model of MES [7]

Obviously, all these activities are not independent. They are integrated into common processes and they share and exchange dynamic information. Furthermore, each MES solution does not integrate every activity and each activity is more or less developed according to the type of industry [13].

## 2.3. The production indicators

In this paper, we focus on the production performance analysis which is, out of the MES activities, the most often implemented one. This activity allows, among others, to provide production indicators to workers and plant managers. Out of every available indicators, the *Overall Equipment Effectiveness (OEE)* is an indicator extensively

used in manufacturing. OEE is, at the same time, easy to use, multi point of view and usable on many systems, whether they are automated or not. Proposed in the 80s in the context of the *Total Productive Maintenance (TPM)*, the definition of OEE has known an important evolution and is not unique [11]. In France, this indicator is standardized since 2002 [1], and we use this standard as reference.

The OEE is the product of three indicators: Quality, Performance and Availability. These three indicators are computed from two kinds of information: numbers of pieces and times. The numbers of pieces are the number of good pieces, the total number of manufactured pieces and the theoretical maximum number of pieces.

Times are useful to differentiate between normal shut-downs (closed time or planned shutdown), downtime losses (for self, dependant or unknown reasons), speed loss and quality loss (see figure 2).

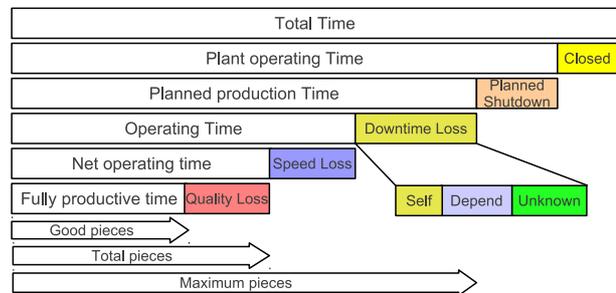


Figure 2. Time portions of total time [1]

Computing and displaying inside the MES the OEE and its parts require to observe occurrences of events related to changes of state of the system. This requires also to observe the quantities and the quality of production. However, all these events cannot be generated by production system control, whatever its degree of automation. Why ?

First, if the quantity of outputs is often easily observable for a discrete production, it is not the case for the quality. The controls are sometimes performed manually or visually, and then the production staff provides the information to the MES. These controls can also be made at the end of the production line or even in a quality control laboratory. So, people who inform the MES are not always the same as those who monitored the production.

The issue is even more significant for states. One must understand that it is very important to detail the downtimes with a maximum number of reasons to allow the staff to react quickly and effectively when the system is badly used, ie when the OEE is low. However, many downtimes used in the OEE computation are not relevant from the point of view of the production system control. For example, when a production system is waiting for a piece, for example to assemble it or transform it, the command does not change its state because it just has to wait for the arrival of the piece to finish the planned opera-

tions. From the control designer's point of view, considering a new specific state would have been a waste of time. From the MES designer's point of view, observing this waiting time is essential for OEE computation in order to record that the station is not really producing in this state. It implies that the production workers have to provide additional information to the events sent to the MES by the production system control.

The measurement of indicators is a classical activity of industrial Manufacturing Execution Systems. So, in these softwares, a data structure exists for this purpose. The MES designer "just" has to adapt this structure to the production system and to synchronize the MES and the production control.

### 3. Proposed Approach

#### 3.1. Requirement Assessment

To pair up a MES with a manufacturing system, one has to model the typical internal structure of a MES application, the internal structure of the target manufacturing system, and the integration of this MES into an industrial domain (featuring an ERP, OPC servers, users...).

The specifications need to be structured so as to make the connection between actors, the tasks to perform and the conditions under which these tasks are done.

For the specific case of OEE, the analysis of the production conditions is important in order to list the possible downtimes of the manufacturing system. This analysis leads to identify the states, corresponding to a breakdown or operating time, and the state transition conditions. These transitions can be caused by the manufacturing system or the production staff, via a user interface of the MES or of the manufacturing system.

Finally, the study is carried out with exchanges of information between all (hardware and software) components that generate the system state changes observed from OEE angle.

#### 3.2. SysML Introduction

The Systems Modeling Language (SysML) is an extension of the Unified Modeling Language (UML), version 2 [5]. While UML is dedicated to software modeling [2], SysML is more general and is dedicated to system modeling [10]. In practice, SysML is well suited for systems with interacting software components and hardware components, such as manufacturing systems, mechanic systems and embedded systems. SysML includes nine diagrams, partially covered by UML (see fig. 3).

#### 3.3. Framework

The development process of the performance analysis function and its pairing up with the manufacturing system is divided into three main steps:

- The analysis of the existing manufacturing system and the customer's specifications which the MES shall answer to;

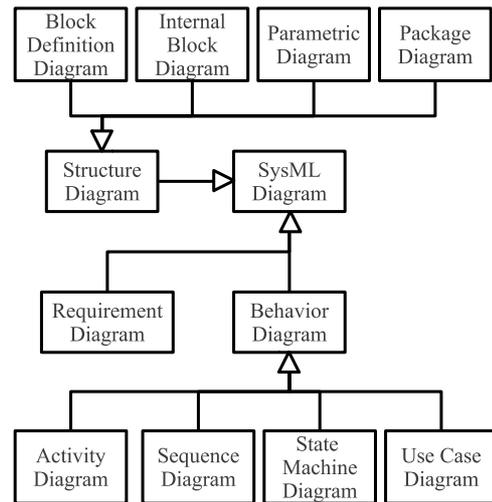


Figure 3. SysML diagram taxonomy

- The MES design with modifications of control programs of the manufacturing system;
- Its implementation with the deployment of the solution and the final validation tests.

The first two steps are based on five different SysML diagrams necessary to specify and design the performance analysis function including the OEE calculation. In detail, the first step of analysis is based on three sub-steps:

1. Analysis of the MES domain to define the environment of the MES, with one Block Definition Diagram (BDD);
2. Analysis of the generic MES structure and existing manufacturing system, with one BDD for the MES and another one for manufacturing system, and several State Machine Diagrams (STM) to analyze production states of each manufacturing system station from a production point of view (but not for a performance analysis);
3. Analysis of customer specifications, with Use Cases Diagrams.

After this first stage of analysis, the second design step is itself split into three sub-stages:

1. Design of the internal structure of the MES, with Internal Block Diagrams (IBD);
2. Design of each MES activity, with various diagrams. For example, for performance analysis, a State Machine Diagram is used to define the different states; it then allows the computation of breakdown and operating times;
3. Pairing up of the MES with the manufacturing system by specification of exchanged information and events through Sequence Diagrams and modifications of initial manufacturing system State Machine Diagrams defined in the first stage.

This approach is applied in next section to a manufacturing system for the performance analysis activity.

## 4. A case study

### 4.1. The Manufacturing System

The studied manufacturing system is an experimental assembly system located in AIP-RAO<sup>1</sup>, a pool of resources and competencies setting industrial equipments at the common disposal of Higher Education Institutes of Rhone-Alps Area (France). Its purpose is to put down colored parts on assemblies conveyed between stations each one on a pallet. It features two supply stations (1 and 2), two production stations (4 and 6) and a supervision station (7). Rescue station (3) and industrial vision based test station (5) are off the subject of this paper. Figure 4 illustrates this experimental system.

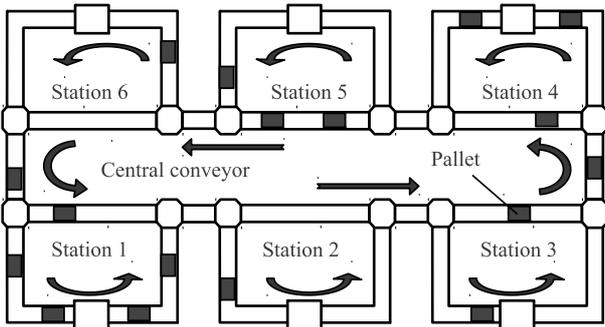


Figure 4. Experimental System

Both stations 1 and 2 enable launching new assemblies and collecting finished ones. Stations 4 and 6 are robotized in order to put colored parts on assemblies. They both feature different stocks of parts (one stock for station 4 and two stocks for station 6). Station 7 coordinates six other ones, monitors functioning states and manages the system energy. The assembly flow is embodied by pallets moving on a conveyor. Each station is controlled by a Programmable Logic Controller (PLC) and is fitted with a basic Human Machine Interface.

The system is designed to be used by a plant manager and one or two workers. The role of the plant manager is, mainly, to plan production lots launching according to weekly production orders received from an Enterprise Resource Planning software, and to make the system produce better by way of OEE analysis. Workers launch new productions, collect finished assemblies, monitor part stocks of stations 4 and 6, and manage system functioning states.

### 4.2. Analysis of Current Situation and Requirements

Due to the format of this paper, we only introduce the time study for OEE computation needs; the quality study is left out. Figure 5 depicts the environment of the system to install. This diagram is characteristic of a classical industrial situation: the MES has to be installed on a pre-existing production system, with an ERP already present. The use of patterns enables to detail the domain and the

other systems to install or adapt; in our case, in addition to the MES, we have to install an OPC server to communicate with PLCs and a database server to store MES data. The production system has to be adapted as it had not been designed to work with a MES.

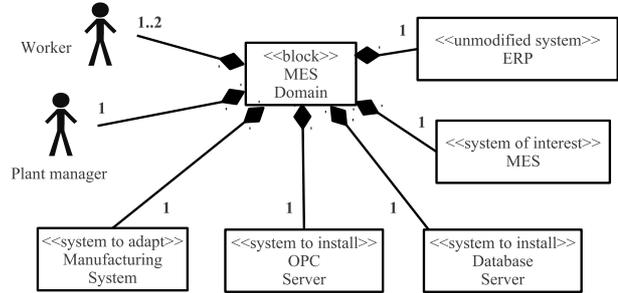


Figure 5. BDD: Global Structure

Figure 6 depicts the manufacturing system architecture. Stations 4 and 6 are the targets of performance measuring. Only PLC programs will have to be modified. Indeed, in the industrial reality, opportunities to modify the physical structure of the system are rare and software modifications are even difficult to perform. It is just matter, most of the time, of information retrieving add-ons, which do not modify the behavior of the system.

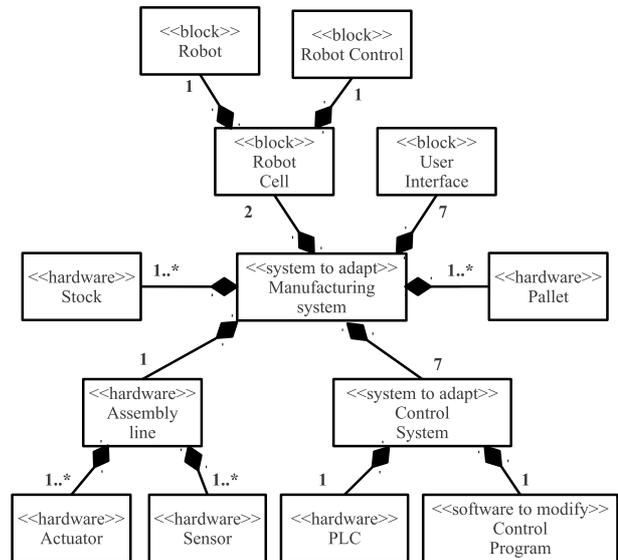


Figure 6. BDD: Manufacturing System

Figure 7 depicts the behavior of station 4 in terms of state changes determined from the SFC program of the station PLC (it is the same for station 6).

The setting up of a MES does not consist in designing from scratch a new tool but rather in configuring/adapting a generic MES to an existing system according to given project requirements. In our case, we appeal to Ordinal Software [12] MES solution Globalscreen. The structure of this MES is founded on the concept of components (see

<sup>1</sup>See <http://aiprao.insa-lyon.fr/>

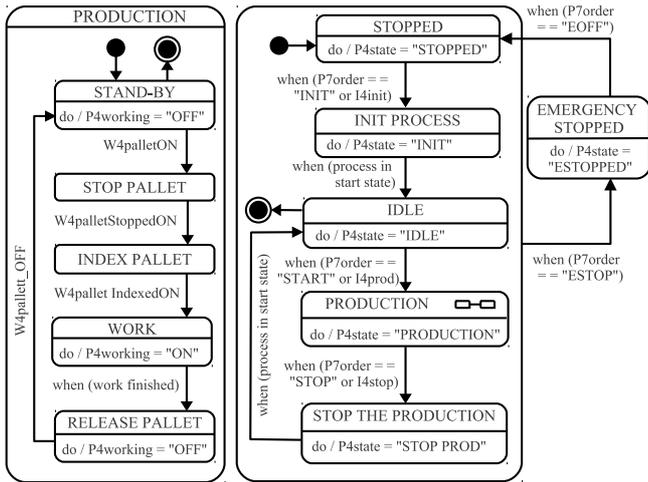


Figure 7. STM: Station 4

fig. 8) featuring a functional part for global variable declaration and their treatment, and visual parts for interaction with users. These components can be structured according to the physical model proposed by ISA [6].

In this model, we limit ourselves to *area*, *process cell* and *unit* levels as higher levels go out of the frame of the workshop and the lower levels (*equipment module* and *control module*) are useless in our case.

Furthermore, our MES can feature OPC clients (only one in our case), *workcenters* (each *workcenter* represents an autonomous functional subsystem inside the MES; we use only one *workcenter* in our case) and a *Graphical User Interface* embodied by *web pages*. These *web pages* are made up of visual part of *component* instances whose *functional part* is located inside the *workcenters*.

Thus, the adaptation of the MES to a given production system consists in designing *components*, instantiating them in *workcenters* and *web pages*.

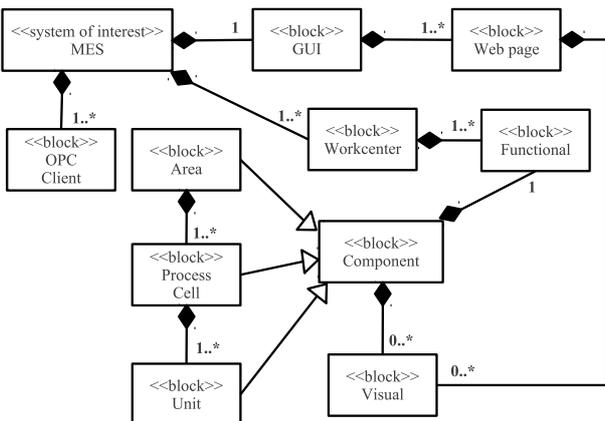


Figure 8. BDD: Manuf. Execution System

For the performance analysis purpose, our MES displays to the plant manager the state of each station (i.e. the value of indicators and, when necessary, the reason of the

current breakdown), prints reports and exports indicators towards the ERP. Workers may need to supplement the information provided by the manufacturing system control, to refine the observed state in the MES. Actually, without the need of any actor intervention, the MES computes indicators over different time ranges: over the day, the week, the month, for instance. Figure 9 depicts these different use cases. Although the production system is present in only one use case, it plays a crucial role as other use cases are of use only if production system state monitoring is correctly performed.

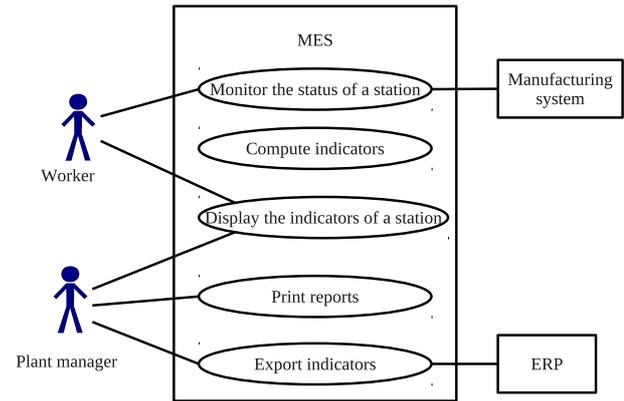


Figure 9. Use Case Diagram

### 4.3. Design

#### 4.3.1. Internal Structure

Figure 10 (on last page of this paper) introduces the instantiation of a part of the blocks previously defined in the Block Definition Diagrams (BDD) 5, 6 and 8. Inside the *Manufacturing System* block instance, the central item is the *PLC4* block instance which represents the station 4 control. *Interface4*, instance of *User Interface* block, can be used by workers to change the functioning state of station 4; this is why they appear here. In the same manner, worker actions on *Interface7* can, by way of *PLC7*, modify the aforementioned state. *PLC4* also interacts with the controller of the robot of station 4.

Figure 10 diagram also features an instance of *OPC Server* block and an instance of *OPC Client* block whereas it does not display any instance of *ERP* nor *Database Server* blocks. As a matter of fact, the instance of *ERP* block exists but it is not used in our example. The instance of *Database Server* also exists and is used for persistence purpose internally by the MES. It does not require any specific study so it is not necessary to display it.

From the point of view of the MES, the central element of the performance analysis is the instance *S4* of a *Process Cell* (see fig. 11). This instance includes the instance *S4-Functional* of block *Functional*, whose role consists in monitoring state changes of station 4. It also includes two instances of *Visual* block: the first one, *S4-indicators* is used to display in real time OEE indicators, different

Time	Time Class
NOT REQUIRED	Closed Planned Shutdown
SETUP	Self Downtime
PRODUCTION	Operating Time
EMERGENCY STOP	Depend Downtime
NO PALLET	Depend Downtime
NO PART	Depend Downtime
ROBOT ERROR	Self Downtime
STOPPED ON SYSTEM	Unknown Downtime
BREAK	Planned Shutdown

Table 1. Station 4 States

downtimes and their evolution. The second one, *S4-States* enables to monitor the changes of state of station 4.

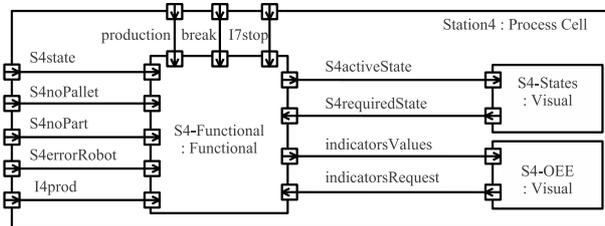


Figure 11. Internal Block Diagram: S4

At this step of design, Internal Block Diagrams are used only for the specification of the structure. Signals exchanged between blocks, although they appear in figures 10 and 11 will only be set through the drawing of State Machine Diagrams which represent the dynamics of each block instance.

#### 4.3.2. Behavior

Before building the State Machine Diagram of *S4-Functional*, one has to study necessary states according to downtimes to observe. On this diagram, a state is created for each time to observe. AFNOR standard provides time definitions which have to be interpreted as time classes. Not every defined time class is used on every production system and, on the contrary, some of these classes may be used several times. Thus, in our system, some classical planned downtime like exchange of tool, periodical setting, quality control or scheduling maintenance are not used. On the other hand (see table 1) the *emergency stop*, *no pallet*, *no part* are depend downtimes. Besides, the times *Closed* and *Planned Shutdown* have been grouped together in the *Not Required* state as it is not necessary to distinguish them for the analysis of the downtimes which could lead to a OEE decreasing.

State changes are depicted in the State Machine Diagram in figure 12. The initial and final states are modeled by the *not required* state. *productionON* event is an event which is not observed on the production system but by

the workers. Indeed, during test, setup and maintenance phases, the production system may have exactly the same behavior as during real production phase. Therefore, it is not possible to decide only from the system observation whether this one is required for production or not. This diagram can be compared with the STM of Station 4 (see fig. 7) to emphasize the differences in state definitions from two different points of view (MES/OEE versus manufacturing system).

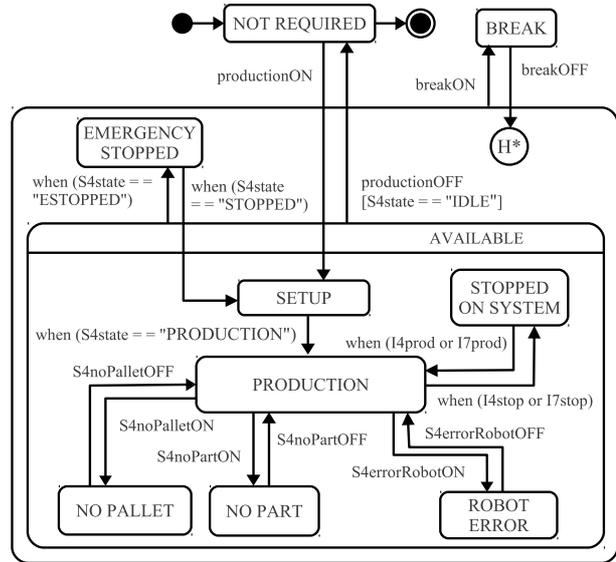


Figure 12. STM: S4 OEE Point of View

In the same manner, *productionOFF*, *breakON* and *breakOFF* events are worker generated events. Notice that *breakON* and *breakOFF* are events which have to be synchronized with the behavior of the production system. Indeed, when workers start their break, the system has to stop producing. On our system, this is done simply by stopping conveyors. Current operations on robotized stations 4 and 6 end up as usual and then every pallet remains stationary: nothing moves any more in the system.

The study of this synchronization between the behavior of instances is performed by way of Sequence Diagrams. For instance, fig 13 (on last page of this paper) shows every interaction necessary to setting up required reactions to a demand of workers to have a break. Adding active states on life lines of instances *PLC4*, *PLC7*, *S4-Functional* and *S7-Functional* helps in highlighting relations between these states. This example highlights the necessity to separate MES OEE states from manufacturing system ones and to precisely model their mutual relations.

The drawing of diagram 12, which depicts different production and downtime states, leads to study the behavior of the manufacturing system from a performance analysis point of view. This study is performed with the help of Sequence and State Machine Diagrams. Setting up state changes induced by users also requires setting up GUI inside the MES. In order to render these state changes (inside the MES) onto the production system, one has to

modify this latter behavior. On our example, it amounts to modifying evolution conditions of station 7. In the same manner every PLC program is supplemented in order to enable data exchanges with the MES, which implies to update initial station STM diagrams such as the one seen in fig. 7. Thus, on diagram 14, bold faced texts inside transitions emphasize the brought modifications of stepping over. In the same manner, actions have to be added to states to inform the MES of the evolution of the production system state. Through this example, we can observe that the study of the evolution of a station may require the study of the behavior of related production equipments.

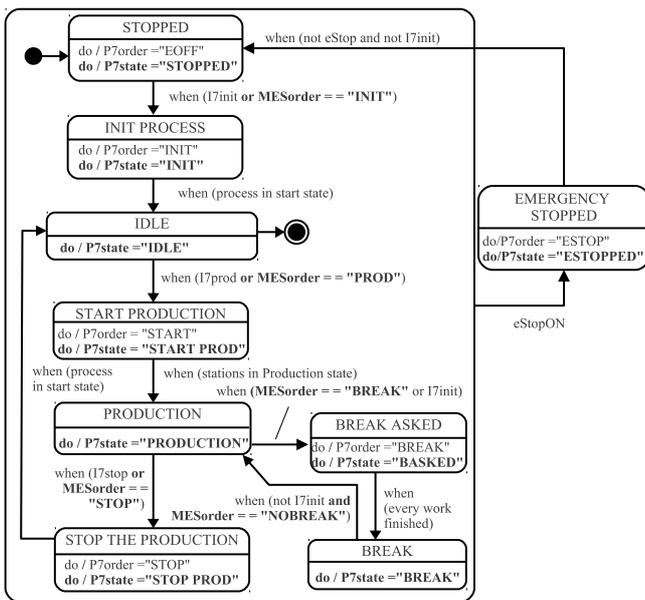


Figure 14. STM: Station 7

It is important to notice that, inside the Sequence Diagrams, we do not look for describing the functioning of information exchange between OPC server and client. It is just matter of specifying exchanges to set up, not the way of carrying them out, as far as OPC technology is standard. In the same manner, we do not study the real-time aspect of the information exchanges between this system components. Indeed, communication capacity of used technology is sufficiently powerful for our use.

Naturally, MES design does not stop at this modeling step. The concrete component making is not a trivial task. Nevertheless, setting-up phase is facilitated by Object Oriented technology which enables to efficiently build classes and their instances from previous diagrams. Moreover, Sequence Diagrams furnish relevant test scenarios according to the making.

The aforementioned example has been made up over a larger set of functions. This would not have been possible without a preliminary detailed design (modeling) phase. Other Sequence Diagrams have been drawn but they could not be included in this paper.

## 5. Conclusion

In this paper we studied the integration of a MES within an existing manufacturing system. We have shown that this integration requires the study of interactions between software components of the MES and software and hardware elements of the manufacturing system. We also showed that SysML is the language well suited for modeling MES paired up with its manufacturing system. In particular, behavior diagrams have a vital role in the study of integration. In our future work, we will endeavor to provide a detailed process of development, analysis from specifications to the detailed design of the MES.

## References

- [1] Association Française de Normalisation (AFNOR). NF E 60-182 : Moyens de production - indicateurs de performances. Taux de rendement synthétique (TRS) - Taux de rendement global (TRG) - Taux de rendement économique (TRE), 2002.
- [2] F. Basile, P. Chiacchio, and D. Del Grosso. Uml-based modeling and model-driven development of distributed control systems. In *13th IEEE intl. conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 1120–1127, 2008.
- [3] P. Blanc, I. Demongodin, and P. Castagna. A holonic approach for manufacturing execution system design: An industrial application. *Engineering Applications of Artificial Intelligence*, 21(3):315 – 330, 2008.
- [4] A. Bratukhin and T. Sauter. Bridging the gap between centralized and distributed manufacturing execution planning. In *Proc. IEEE Conf. Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2010.
- [5] S. Friedenthal, A. Moore, and R. Steiner. *A practical guide to SysML – The systems modeling language*. The MK/OMG Press. Elsevier, 2009.
- [6] International Society of Automation. ANSI/ISA-88.01. Batch control part 1: Models and terminology, 1995.
- [7] International Society of Automation. ANSI/ISA-95.00.03. Enterprise - control system integration. Part 3: activity models of manufacturing operations management, 2005.
- [8] International Society of Automation. ANSI/ISA-95.00.01. Enterprise - control system integration. Part 1: models and terminology, 2010.
- [9] S. Iwatsu, Y. Watanabe, and K. Kodama. Development of communications protocols between manufacturing execution system and production equipment. In *12th IEEE intl. conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 280–287, 2007.
- [10] M. Linhares, R. de Oliveira, J.-M. Farines, and F. Vernadat. Introducing the modeling and verification process in sysml. In *12th IEEE intl. conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 344–351, 2007.
- [11] P. Muchiri and L. Pintelon. Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion. *International Journal of Production Research*, 46:3517–3535, 2008.
- [12] Ordinal Software. A robust and innovating platform: the COOX platform, 2011.
- [13] J. Snoeij. MES Product Survey. Technical report, Logica, 2006.

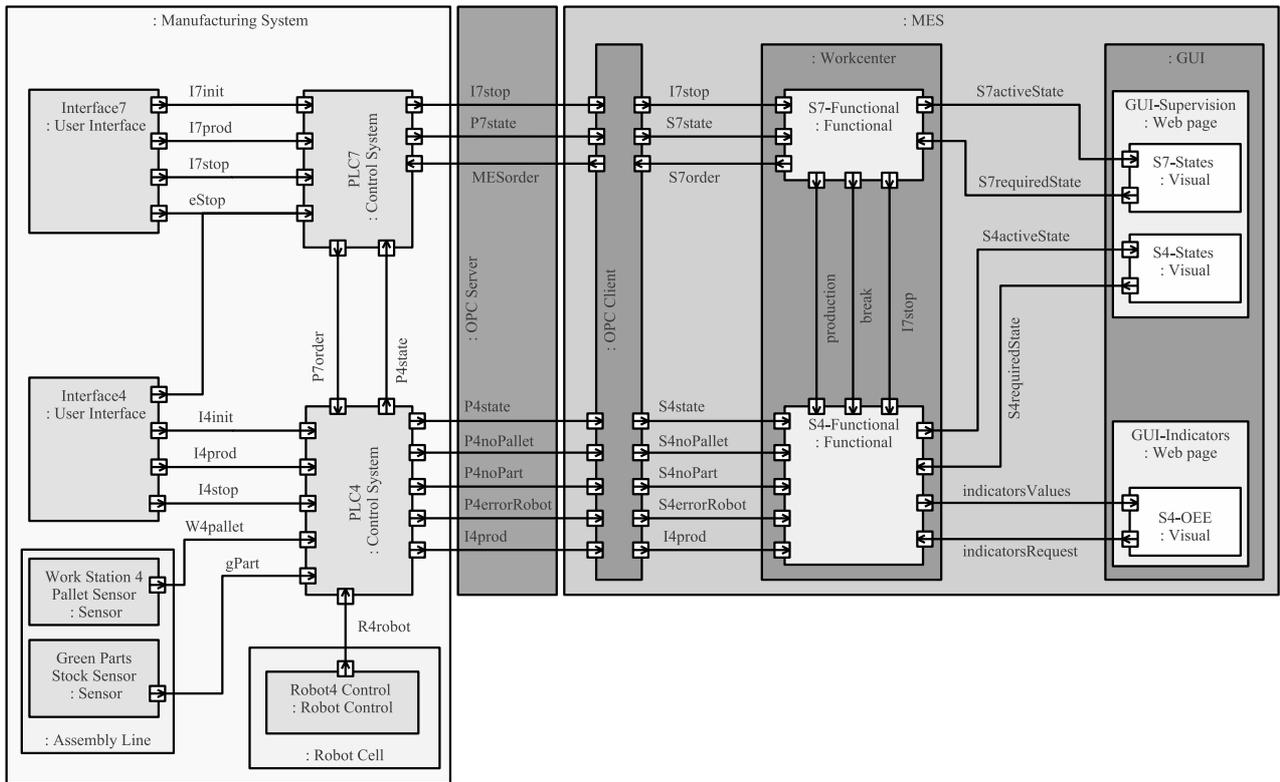


Figure 10. Internal Block Diagram

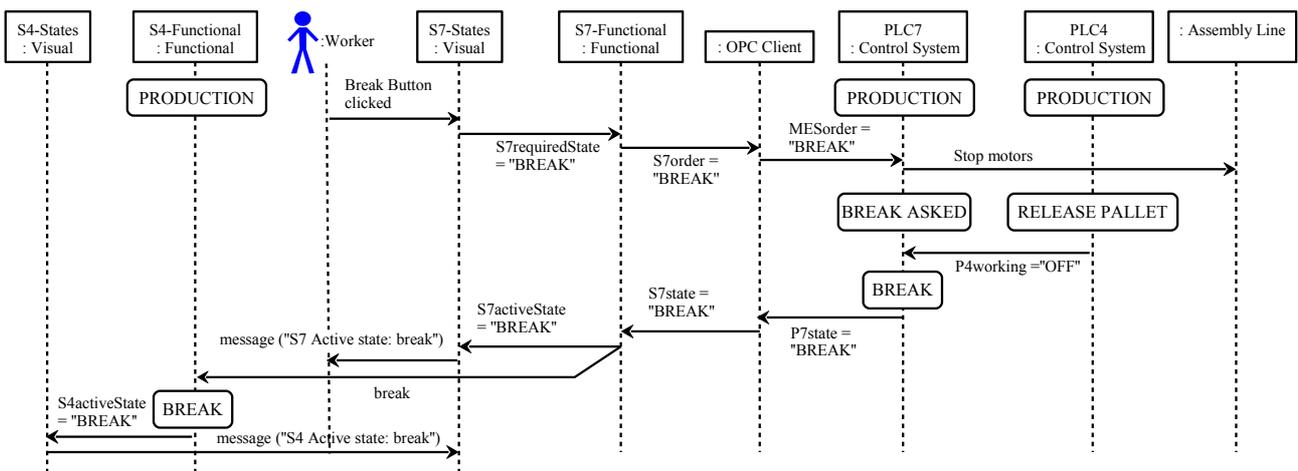


Figure 13. SD: Having a Break