

# A Systematic approach of Multi Agents in FMS for Plant Automation as per the future requirements

Rahul Vyas<sup>#</sup>, Chandra Kishan Bissa<sup>#</sup>, Dr.Dinesh Shringi\*, Dr. Kamlesh Purohit<sup>§</sup>

<sup>#</sup>P.G Student M.E., Department of mechanical engineering, MBMEC, JNVU, Jodhpur, Rajasthan

\* Associate Professor, Department of mechanical engineering, MBMEC, JNVU, Jodhpur, Rajasthan

<sup>§</sup>Professor & HOD, Department of mechanical engineering, MBMEC, JNVU, Jodhpur, Rajasthan

thebissaji@gmail.com

rahulchi.vyas@gmail.com

drkpurohit@gmail.com

**Abstract**—Today Flexible manufacturing system is found to be more promising technology for mass production of the products as it consists flexible installations of workstation which are always ready to be changed as per the new products and demand. This is a very well known technology having a group of processing workstations interconnected by means of an automated material handling & storage system & controlled by integrated computer control system that is why the name given to it is 'flexible'. Flexibility has become a key factor for manufacturing to keep competitive. Software agent technology can be widely used to improve the flexibility of a plant and the Manufacturing Execution System (MES) used to control production. This paper introduces a multi-agent system design consisting of four types of agents covering the three layers of the plant automation pyramid from Enterprise Resource Planning (ERP), Manufacturing Execution System (MES), to the field control layer. The architecture and the communication protocols of each agent are presented and the increase in flexibility for mass customized and highly dynamically changing products such as car manufacturing or manifold production are given. This paper is structured as the following: First the PABADIS' PROMISE system architecture is introduced. Then the architecture of each agent and the PABADIS' PROMISE MAS are presented. Finally, the communication protocols between the PABADIS' PROMISE agents are described.

**Keywords**- Flexible manufacturing system, Manufacturing Execution System (MES), Enterprise Resource Planning(ERP).

## I. INTRODUCTION

As [1] and [2] mentioned, manufacturing process development can be divided into three periods: pre computer numerical control, computer numerical control(CNC), and knowledge epochs. In the pre-CNC epochs, the market was characterized by local competition and there were small demands for product variations. Manufacturing put focus on increasing production rate. In the CNC epoch, the emphasis was changed to cost reduction and product quality control. In the knowledge epoch, with intensified global competition and improvement of computer and information technology, manufacturing is required to be able to response rapidly to a

fluctuating market. Flexibility is a key factor to maintain the competitiveness of a plant. As defined in [3], flexibility is the "ability to better meet customer needs by modifying existing products". Conventional centralized manufacturing models show big deficiencies to satisfy frequently changing requirements introduced by markets. Enterprises are therefore adopting advanced models and technologies characterized as distributed, collaborative and flexible [4] such as software agents. Software agents are very attractive due to their native properties of autonomy, communication, coordination, and reaction [5, 6]. Exemplarily ADDYMS (Architecture for Distributed Dynamic Manufacturing Scheduling) [7], AIMS (Agile Infrastructure for Manufacturing System) [8], MetaMorph I [9], MetaMorph II [10], and PABADIS (Plant Automation Based on Distributed System) [11] can be named as representatives for multi-agent systems (MAS) used to introduce flexibility to manufacturing.

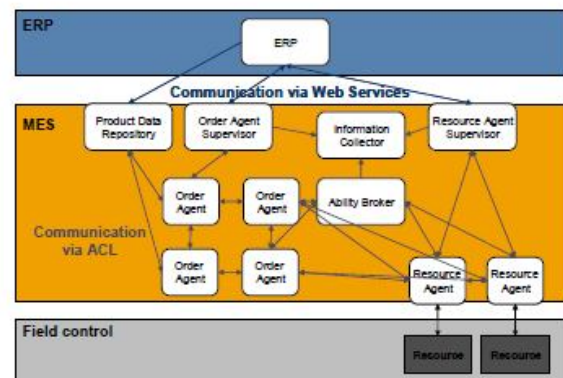


Fig. 1. PABADIS' PROMISE automation pyramid

Based on the results of EU project PABADIS, EU funded project PABADIS based Product Oriented Manufacturing Systems for Reconfigurable Enterprises (PABADIS' PROMISE) advances the use of multi agent

systems (MAS) to allow for the new paradigm of “the order is the application” [12]. The PABADIS’PROMISE MAS covers all three levels of plant automation pyramid: ERP, MES, and field control devices. This paper will concentrate on the description of individual agents in the PABADIS’PROMISE MAS as well as the communication protocols.

## II. PABADIS’PROMISE SYSTEM ARCHITECTURE

The PABADIS’PROMISE architecture [5] covers all three layers of the plant automation pyramid as shown in Figure 1. The novelty of the approach in PABADIS’PROMISE is to gain flexibility by shifting the decision-making process from the ERP system down to the MES layer.

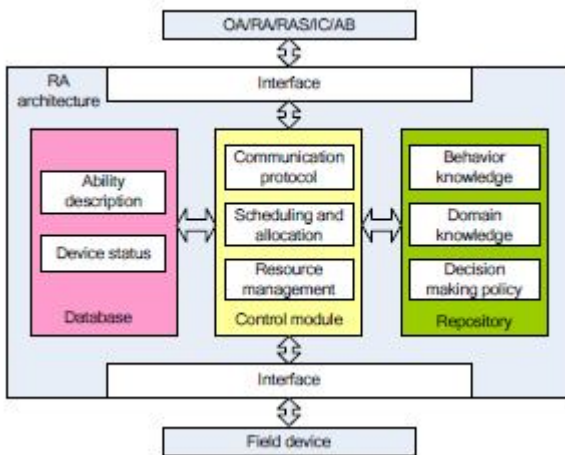


Fig. 2. RA architecture

The technology to provide this flexibility in the MES is an agent based system described in this article. Furthermore, the field control layer is integrated into the architecture by a set of plug-and-participate mechanisms that allow for flexibility and adaptability of the shop floor.

To fulfill this task the PABADIS’PROMISE architecture consists of eight components (Figure 1):

- Enterprise Resource Planning (ERP) system: responsible for the order management at ERP level and interface to the super ordinate layers of a company and the customer.
- Order Agent Supervisor (OAS): responsible for the Creation and initialization of order agents and for the Supervisory control of their life cycles.
- Order Agent (OA): responsible for the execution control of a production order.
- Resource Agent (RA): responsible for the management of a resource.
- Resource Agent Supervisor (RAS): responsible for supervision and control of resources. It can be contacted by ERP or other applications such as SCADA to monitor, configure, control maintenance and manage resources.

- Information Collector (IC): responsible for the collection and provision of the order and resource related data.
- Ability Broker (AB): responsible to make all resource abilities and resource availability known in the system. It provides facilities to collect abilities announced by a resource manager and a look-up service to query for certain abilities.
- Product Data Repository (PDR): maintains a database of bill of material. It is the “cookbook” of the system. The core of the PABADIS’PROMISE architecture is the MES layer that is represented by the MAS and has two primary goals: 1) to execute the ERP orders and 2) to control the field layer devices. These general goals come from the basic purpose of the MES – to provide a link between the customer (at the ERP) and the resource (field). Therefore, the MAS in PABADIS’PROMISE can be divided into three main categories:
  - ERP interface (OAS, RAS)
  - MES kernel (OA)
  - Field control interface (RA)

Such components as PDR, IC and AB are the database support and are not discussed in the next chapter that describes the architecture of the four main agents.

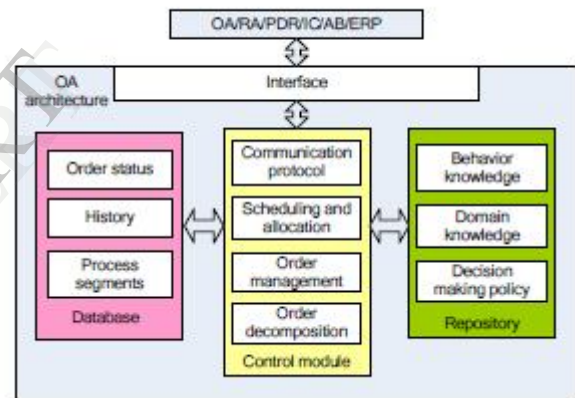


Fig. 3. OA architecture

## III. PABADIS’PROMISE MAS ARCHITECTURE AND AGENT ARCHITECTURE

According to the categories in [14] and the functionalities of each agent, the OA and RA are designed as a hybrid collaborative and deliberate agent. OAS and RAS do not need to take complex decisions and mainly monitor and forward control commands to OA and RA agents. They are therefore designed as reactive agents.

**Resource Agent (RA):-** The RA is responsible for the overall management and operation of a resource. As a deliberate agent, the RA makes its decision based on the belief-desire-intention (BDI) model. Based on the general architecture of BDI agent [15], the architecture of the RA is shown in Fig. 2.

The RA is the link between the MAS and the field device controlled by it. It therefore consists of two interfaces: a standardized FIPA interface to the agent system and a resource specific interface to the field level to retrieve and sent information to the controlled device. At start-up the RA initializes the resource and registers its abilities in the AB. Also when there are changes of its own abilities and availability, the RA is responsible to inform the AB about changes. During operational state the RA is responsible for processing production requests from OAs. It therefore needs to schedule and allocate its underlying resource and to supervise exception in the execution of the allocated processes on the resource. In case of exceptions the RA also must inform OAs booked on it to reschedule their work flow. The repositories of the RA support the above actions. Especially the behavior knowledge that allows the RA to efficiently react on exceptions and perform alternative actions and the domain knowledge required for scheduling decisions must be mentioned here. Details on scheduling in PABADIS' PROMISE can be found in [16].

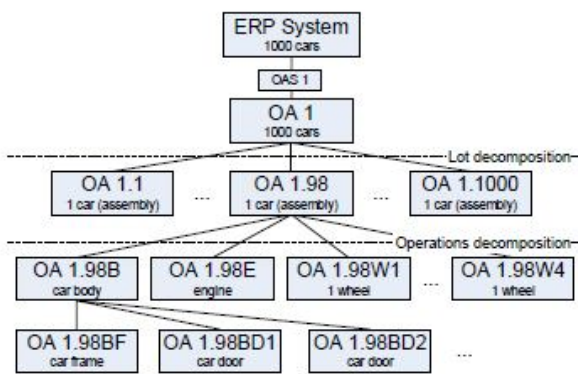


Fig. 4. Order decomposition

**B) Order Agent. (OA)** The Order Agent (OA) is the key component of the PABADIS' PROMISE concept that autonomously organizes and executes a production order given by the ERP system. Following the paradigm "The order is the application" an OA is responsible for one directly assigned product. In opposite to centralized approaches where the complete production is planned, the OA only takes decisions for its on product. The main goal of this approach is not to reach a total optimum in production rather, but the aim is to react in short time to changes in the order flow, e.g., late order freeze, or on the shop floor such as maintenance or machine failures. In order to improve utilization of the plant, OA show a benevolent behavior especially during re-scheduling. The architecture of OA is shown in Fig. 3. It is quite similar to RA's due to the fact that both RA and OA are deliberate agents. The repositories form the base for all decision making and the database holds production status and all relevant data to execute the process segments. The functionality an OA has to offer range from order decomposition, i.e. splitting a lot into products and subcomponents, to the management and execution of the process segments needed to fulfill an order to scheduling of process segments and allocation of required resources.

Depending on the complexity of a production order an OA can create further OAs to fulfill subtasks of its production order. I.e., simple orders will be completely done by a single OA whereas complex orders such as the production of a car usually decompose into multiple levels of OAs (see Fig. 4). The following three flavors of an OA can be identified: lotOA, productOA, and processOA: The lotOA is only required for bigger production lots. Its main functionality is to start and supervise the production of the lot and therefore only implements functionality for lot decomposition and limited scheduling. *ProductOAs* do the main workload in producing and assembling the final products they are responsible for. Their functionality includes order management and scheduling as well as further decomposition of products. Finally, *processOAs* are responsible for fulfilling the production of subcomponents, and include functionality for resource allocation and fine grained planning. The depth of operation decomposition is not limited by the system, but naturally will be limited by organization of production and the overhead introduced by agent creation and agent-to-agent communication and hence functionalities assigned to a *lotOA*, *product OA* or *process OA* might be implemented in a single agent instance.

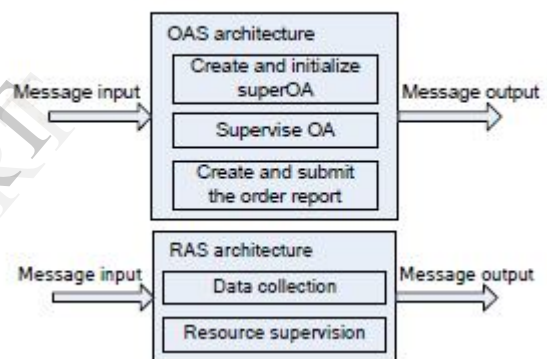


Fig. 5. Order and Resource Agent Supervisors architecture

Hierarchical decomposition of an order and instantiation of the respective agents bring the following advantages:

- An individual agent does not become too complex and is easy to develop.

- The code especially of a processOA is simple and small decreasing the requirements for resource limited devices such as RFID or embedded platforms.

Especially active RFID tags hosting OAs are of importance to the PABADIS' PROMISE concept since they allow physical association of product and responsible order agent. Favors of this approach are easy product identification and location and storage of product relevant data and a responsible agent together with the product that helps to reduce overheads in highly dynamical environment. Detailed information about the combination of agent and RFID and their advantages would go far beyond the scope of this paper and can be found in [16].



### C. Order and Resource Agent Supervisor (OAS, RAS)

The Order Agent Supervisor is an interface between ERP and MES. It has the following functionalities:

- Instantiation of ERP orders: An OAS receives production orders from the ERP and initializes them by creating appropriate OAs. Depending on the order complexity and the decomposition strategy, either a *lotOA* or directly a *productOA* is applied.
- Supervision of production orders: The OAS gets the reports about the order execution from its OAs and forwards them to the ERP system. The OAS also reacts on status requests or order changes from the ERP and accordingly supervises the OAs. To find a compromise between granularity and overhead within PABADIS'PROMISE, a milestone based reporting is introduced. E.g. a customer might not be interested if a certain screw is fixed, rather he wants to know if his car is painted red or if he can still change the color. The implementation of the OAS is a reactive agent that contains several simple, parallel-organized competence modules. Based on the general architecture of a reactive agent described in [7], the architecture of the OAS is shown in Fig. 5. The purpose of having the RAS is to provide additional access points to the shop floor of the system apart from the ERP – OAS – OA – RA mechanism. These mechanisms can be used for different purposes such as coordination of the resources or integration of legacy systems for data collection or influencing the behavior of a resource such as SCADA or operator control stations. Similar to the OAS, the RAS is also a reactive agent. The architecture of RA is shown in Fig. 5. The Resource supervision module also provides a possibility for a human machine interface. The Data collection module is responsible for collecting necessary data for human supervision and resource management.

### IV FLEXIBILITY BY COOPERATION

The approach taken in PABADIS'PROMISE is to maintain necessary flexibility required in future manufacturing by introducing cooperation between agents. A single agent has limited knowledge and resources. For a complex task, several agents need to coordinate with each other to find a solution. The multi-agent system (MAS) of PABADIS'PROMISE provides a communication (cooperation) infrastructure to allow the four main agents to fulfill their tasks. Agents are organized in the hierarchical way (Fig 1 and Fig 4) but collaboration between different hierarchy levels is allowed. The OAS first receives the production order from the ERP. It creates one or multiple OAs corresponding to the order. According to its production data, missing information is complemented from the PDR. Then the OAs query for needed abilities at RAs using the AB and finally set up a schedule for their "private" production. (advanced) flexibility introduced by the MAS on two independent aspects: first is *order flow management*. Since an OA is only responsible for its own product and also for organizing the production scheduling

for this product, changes can be introduced at any time. In this respect changes can be performed as long as a production step is not executed or previous completed steps prevent the change. Changes will not influence other products since the scheduling is also "local" to one agent and dynamically created based on the latest status of resources. The second aspect is *resource management*. Due to the fact that there are no fixed relations between OAs and RAs, failed resources can be dynamically replaced, therefore, easing maintenance, change of equipment etc. Also load-distribution among multiple resources of the same kind is done automatically since OAs request an ability and not a certain resource. The authors are aware that optimization is a crucial aspect since a flexible yet purely utilized shop floor is not acceptable. Ongoing investigations show promising approaches that allow combining good optimization together with higher flexibility due to the simpler system concept. Investigated issues are ERP design that is ability and not resource centric and local scheduling domains to reduce communication effort. Agent systems from other domains indicate that the above goals can be reached for highly customized products and highly dynamic shop floors. The PABADIS'PROMISE project therefore installs a field test to verify the expected results.

### V COMMUNICATIONS IN PABADIS'PROMISE MAS

Agents in the PABADIS'PROMISE MAS have to coordinate with each other to fulfill their production tasks. The successful coordination is based on an efficient and effective communication. In the following sections, the proposed agent-centric approach is described in detail and the communication protocols are described.

#### 1) RA Communications

Fig. 7 is the sequence diagram of the communication links of the RA. Via the communications with the AB, the RA can register and change its abilities and availability. The RA can also look up needed abilities in the AB. Additionally the RAS can be informed to record and notify Super ordinate systems about ability changes and exceptions. The RA can also communicate with other RAs for scheduling as well as with OAs as shown in the Fig 6, depending on the different mechanisms for scheduling and resource allocation.

#### 2) OA Communications:-

OAs are key components of the MES layer in a PABADIS'PROMISE system. As described before, there are three types of OAs in PABADIS'PROMISE system. The lot OA receives the production order from OAS. Both Lot OA and product OA need to contact PDR to collect the needed information to decompose an order and plan the execution. A product OA creates one or more process OA(s). A process OA then negotiates with RA(s) for scheduling and resource allocation based on the information from the AB. After the production finishes, the process OA sends a report to its product OA, that creates a report to its lot OA

and the lot OA submits a collected report to OAS. In the Fig. 6, the communications of all three types of OAs are Shown . The horizontal communication between product OAs is not included.

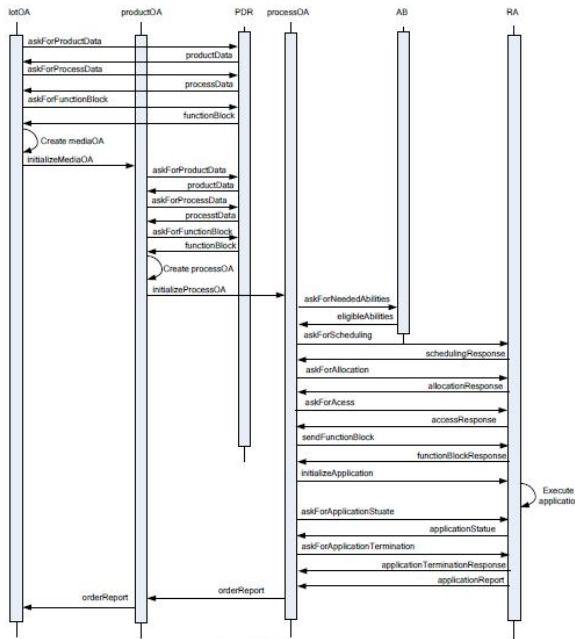


Fig. 6. OA communications

**C. OAS and RAS Communications**

Communication patterns of the OAS and the RAS are given in Fig. 8 and Fig. 9. After receiving a production order the OAS creates a lot OA. When the production order is finished, the OAS receives a report from the lot OA and creates its report to the ERP. The RAS provides interfaces to the legacy systems such as HMI as well as providing an interface to the ERP to control the shop floor. The RAS can influence a resource by changing the current schedule, current state, features, and supervises utilization of a resource, which are all included in supervise commands. The RAS communications are shown in Fig. 9. Additionally, the RAS is involved in the resource discovery mechanism, which is triggered either by the RA or by the ERP, where the RAS connects two entities to synchrosize respective local databases.

the decentralized and hierarchical control method the execution of process or the plan which is formed by production management will be completed in a proper channel [14]. With the help of above figure we came to know how it works. First the production management considering all the factors , planned to manufacture number of products [4]. Now this information is send to the central control system. The central control system appoints the supervisor for each and every function separately, which is in contact with central supervisor and follows all the instructions given to them. As shown the transportation supervisor [3] has assigned with the task how the raw materials flow from one workstation to another workstation .A signal or message

should be convey to them by central control station regarding to the materials status. Once the transportation supervisor completed it tasks it informs back with the help of signals to the central controller system. In the same way all the supervisors completed their task and report directly to the central controller. As the control framework is hierarchical there is no communication between two supervisors.

**VI. A METHODOLOGY TO INTEGRATE A LOGICAL MODEL AND PHYSICAL MODEL**

Our goal in this paper is to propose a Methodology which allows us to model the control in a FMS within a logical layer, where as the material part is modeled within a physical layer [16].To do so each job is associated by the entity known as physical entity, and for each physical entity there is a corresponding logical entity. The logical entity [5] flows between the supervisors while the physical entity flows between the facilities. Logical and physical entities are synchronized using signal and wait functions process which helps us to block an entity (physical or logical) in a transaction waiting for its corresponding entity to send a signal. For example process is hold on a machine because it is blocked and waiting for the signal which is send to it by corresponding logical entity . Once it receives the signals it transformed or undergoes machining process. After that logical entity will wait for the signal that provided by the physical entity for performing the particular function on it and this signals through logical entity send back to central control station. For implementing the modeling methodology [6] the software chosen was SIMAN / ARENA and this tool also consists specific simulation techniques devoted to representation of manufacturing systems.

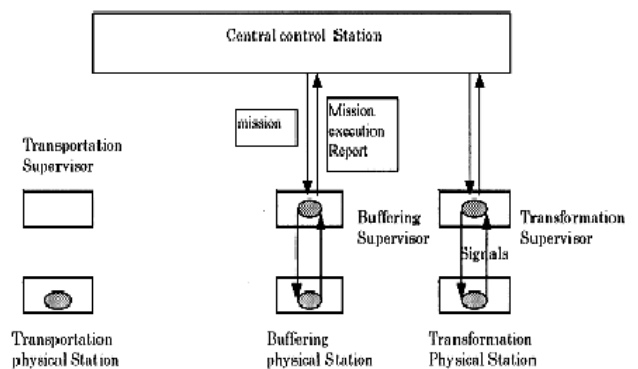


Figure 2 Logical and physical layers

In the complete model (i.e., the model involving a logical layer and a physical layer) logical entities are used:

- To model the arrival of work orders at the central control station.
- To model the transmission of signals between the central control station and local supervisors.
- To send signals of beginning operation to the corresponding physical entities, in order to activate them. As a matter of fact, a physical entity waits in a queue until this signal of "begin operation" arrives. Due to the fact that Arena requires a resource that has to be allocated to the entity which uses it, the resources reservation are done by the physical entity in the same way as that for the physical model. When the operation on the allocation resources is executed, the physical entity sends the signal of "end of operation" to the corresponding logical entity which is waiting for this signal in a queue.
- To model the transmission of "operation execution report" message between the local supervisor and the central control system.

Hence, control modeling allows to model:

- Synchronization mechanism between logical and physical entities.
- Transmission times for signals or messages exchanged between central and local supervisors...

## VII. EXAMPLE OF FMS

### A. The FMS model

The example presented in this paper is a type of multi-machines FMS (MMFMS) as defined by MacCarthy and Liu [11]. For our purpose, we use two items defined by MacCarthy a single flexible machine (SFM) is a computer controlled production unit with tool changing capability, a hardware handling and a part storage buffer. An MMFMS is a type of FMS with several SFMs connected by a material handling system which is capable of visiting two or more machines at a time.

When a transformation order arrives into the shop, a new job is made by taking an empty pallet and carrying in it the parts to be processed. This operation is made in the manual workshop. At the end of this operation, the job (i.e., the pallet with the parts) is stored in the automated buffering unit using one of the load/unload tables. The routing of a job contains two or three transformation phases. The following sequence of operations describes a transformation phase. If there is an SFM available and a job waiting in the stock, this SFM is reserved for this job by the transformation supervisor; a free transport unit (e.g., an AGV) goes to the shop store and the waiting job is removed from the stock. Then it is deposited to the transport unit when this one arrives at the buffering station. The vehicle transports the job to be

processed to the reserved SFM. When the transformation operation is achieved, in the same way, the job returns to the stock using a transport unit.

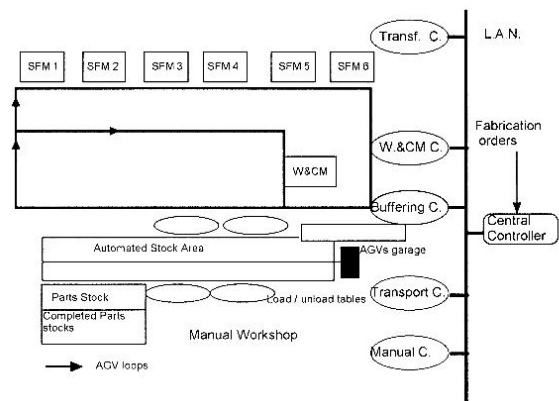


Figure 3 Example of FMS

Then it waits for the washing and measures control station to be available. When this station is idle, washing and measures control supervisor reserves it for this job and the job goes to the control station using a transport unit. At the end of the measures control, the job goes back to the stock. Execution of a part is composed of two or three transformation phases: between two transformations phases a part has to be removed and fixed in another position on its pallet. This work is done in the manual workshop. Due to the fact that between each operation (transformation phase or manual operation) a job always returns to the stock, operations can be considered as desynchronized. Thus, the automated part of the FMS could work without an operator being present in the FMS.

### B. Control modelling

The facilities of this FMS example are represented by five different stations: a station for the manual workshop, a station for the automated buffering store, a station for the single flexible machines (SFMs), a station for the washing and measures control unit and a station which deals with the transportation (Fig. 3). To model the control, one central control station (i.e. central supervisor) is defined with five control supervisors: the manual workshop supervisor which receives the carrying or removing missions realized by one of the operators. Whenever a worker ends an operation, he reports it to the manual workshop supervisor via a computer, the automated buffering supervisor which controls the inputs and outputs of empty pallets or pallets with parts (jobs), the single supervisor for all the SFMs which controls the schedule of each SFM by allocating waiting jobs to a free SFM. The allocated jobs are chosen by the central control supervisor which is in charge of the schedule of all the shop, but the SFM supervisor will choose the appropriate SFM. The washing and measures control supervisor which controls the washing and measures control operations, the transportation supervisor which controls the schedule of each

transport unit by allocating to it a waiting pallet. The choice of the pallet is realized by the central control supervisor but the transportation supervisor is in charge of the choice of the transport unit which will execute this mission. For this, the transportation supervisor assigns the destination (either the SFM or the washing and measures control unit) to the chosen transport unit. The central control supervisor coordinates these five supervisors and communicates with the production management module. We made the assumption that the control is made using pieces of software in each computer representing a local supervisor. The data exchange (i.e., transmission of missions and transmission of reports) are made using manufacturing message specification (MMS) on an industrial Ethernet local area network [8-9]. The data exchange is modelled by a point-to point bi-directional link between the central control supervisor and each local supervisor (there is no exchange between two local supervisors). It means that the transmission of an order from the central control supervisor to a local supervisor is always followed by the transmission of the execution report message from this supervisor to the central control supervisor. Due to the use of MMS software, the messages are always well routed and only the average transmission time of a message has been taken into account in our simulation model [12].

## VIII. RESULTS OF ANALYSIS

We are interested in the incidence of the control on the proper execution of the operations in the shop. This control and the way it is organized (i.e., decentralized and hierarchical in our example) [10] can generate some delay on the execution of the operations. Another factor which affects the performance of the control is the physical characteristics of the network: in our case, the transmission speed. In order to evaluate the incidence of the control layer on the shop performance, two simulation models were realized. One describing only the routings and physical objects (vehicles, machines, operators and stocks) called the 'physical model' and the other, called the 'complete model', describes the physical layout (i.e., resources, routings and stocks) plus the control layout (i.e., messages, report execution, logical entities, local and central supervisors).

We measure the mean flow time, which is the mean time spent by a job in the shop on these two simulation models. In our example, FMS produces an aggregate mix of six types of parts in equal numbers (i.e., for an FMS production of 6 jobs, one part of each type is produced). Parts type 1-type 5 have production route-sheet with two transformation phases, and the type 6 parts have production route-sheet with three transformation phases. The time process for a phase varies from 30 to 60 minutes. All the jobs enter the shop at the beginning of the simulation. This implies that a low production gives low shop congestion and a high production demand gives high level shop congestion. The priority rules

chosen are the following: parts are fixed on pallets with a first in first out priority. As the parts enter the system depending on their type (firstly, the type 1 parts, then the type 2 parts, and so on), it means that parts waiting to be "fixed on pallets are scheduled depending on their type (1-6). In order to minimize the time spent in system, jobs waiting for a resource are scheduled with the priority rule fewest remaining operations (FOPN), which gives priority to the job with the fewest remaining operations.

To produce a type 1-type 5 part (with only two transformation phases) the central supervisor has to send 27 mission orders to local supervisors. So the data control flow for the production is composed of 54 messages (mission orders and mission execution reports). To produce a type 6 part (with three transformation phases), the data flow control is composed of 78 messages. We assume that the messages exchange rate is 30 messages per second. Thus control modelling adds 2 or 3 seconds (for messages transmission) to the lead time (time which is necessary to obtain a completed part). When we first simulated the physical model (i.e. the model without the control layout) versus the complete model (i.e., the model including the physical) layout as well as the control layout), we obtained results which present a strong variation between the two models. These variations were due to the fact that the physical entities were queued according to the selected priority rule.

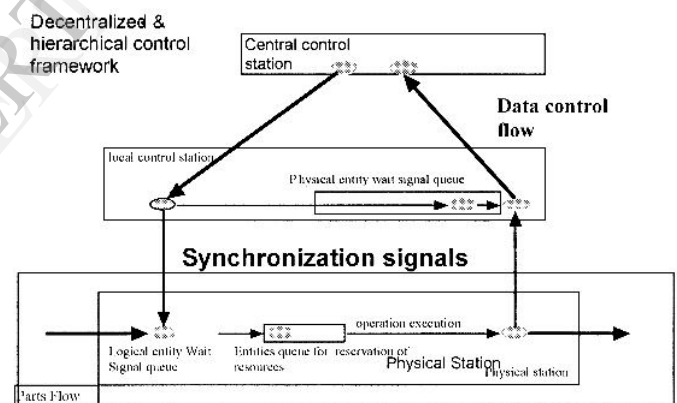


Figure 4 Synchronising and reversing resources mechanism in the complete model.

(e.g. in our case, the FOPN rule) whereas the logical entities were not queued and were arriving in the same order than they have been sent (i.e., according to the FIFO priority rule). This was due to the synchronization mechanism used in Arena, i.e., the Signal/Wait procedure, which is not adapted for scheduling. To correct this problem, we have introduced for each allocation resource two waiting queues for the physical entities. One, which is scheduled using FIFO, is only used to model the synchronization mechanism between the logical layer and the physical layer. The second queue uses a priority rule and allows us to select the appropriate



physical entity depending on the selected scheduling strategy (Fig. 4).

Table 1

Flow time variations between physical and complete models depending on the number of completed jobs

Production shop (jobs)	Mean flowtimes for physical model (min)	Mean flowtimes for complete model (min)	Flow tome variations (%)
6 jobs	323	317	1.9
60 jobs	1268	1264	0.3
120 jobs	2341	2327	0.6
180 jobs	3582	3553	0.8
240 jobs	4714	4702	0.3
300 jobs	5743	5729	0.2
360 jobs	7031	7004	0.4

The results obtained with this new scheduling modelling (Table 1) show that the flow time between the two models (i.e. the complete one and the physical one) are quite similar. Variations are not significant (below 1%) and no trend seems to appear in these results (i.e., the variations does not seem to depend on the number of jobs in the system). However, this work does not take into account the times of process control which can be necessary to establish some control decisions. Moreover, due to our modelling methodology, we were not able to model the allocation of resources throughout the local supervisors and the logical entities, in our model the allocation of resources is always realized through physical entities. These results have to be validated by further experiments: in particular, other performance measures such as the transportation time or the utilization rates of resources and their evolution with regard to the number of transport units, the number of SFMs, the operating times, the network communications in the shop, the schedules of the workers (e.g., one, two or three shifts) should be evaluated.

#### IX. CONCLUSION AND PERSPECTIVES

We have proposed a methodology which allows the integration in a single-simulation model, a physical model which corresponds to the material elements of the FMS with their physical characteristics and interactions, and a logical model which corresponds to the modelling of the computer control system and its interaction with the hardware part. This integration allows us to evaluate the incidence of the control framework on the performance of the production system. It also allows us to test different control frameworks on an existing manufacturing system and to select the one that is the more adapted with regard to the physical constraints and the production objectives. One of our perspectives is to model the allocation resources throughout the local supervisors and the logical entities. Other

perspectives are: to integrate the dynamic scheduling rules and to test the way these dynamic strategies can be implemented in a control framework:

- a) To use this integrated model simulation to establish procedures in case of a degraded utilization of the elements of the shop,
- b) to use this integrated model simulation to establish procedures in case of a degraded utilization of the elements of the shop

#### X. REFERENCES

- [1] Albert D Baker, A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull , Journal of Manufacturing Systems, Research 17 (1998) 297-320.
- [2] Babak Shirazi, Iraj Mahdavi, Nezam Mahdavi-Amiri, iCoSim-FMS: An intelligent co-simulator for the adaptive control of complex flexible manufacturing systems, Simulation Modelling Practice and Theory, Research 19(2011) 1668-1688.
- [3] B. Malakooti, A hierarchical, multi-objective approach to the analysis, design, and selection of computer-integrated manufacturing systems, Robotics and Computer-Integrated Manufacturing, (1989) 83-97.
- [4] D. Veeramani, B. Bhargava, M.M. Barash, Information system architecture for heterarchical control of large FMSs, Computer Integrated Manufacturing Systems, Research 6(1993) 76-92.
- [5] F.J.A.M. van Houten, Manufacturing Interfaces, CIRP Annals - Manufacturing Technology, Research 2 (1992)699-710.
- [6] H.-P. Wiendahl, H.A. ElMaraghy, P. Nyhuis, M.F. Zäh, H.-H. Wiendahl, N. Duffie, M. Brieke, Changeable Manufacturing - Classification, Design and Operation , CIRP Annals - Manufacturing Technology, Research ( 56,) 2007 83-809.
- [7] Han, McGinnis, Flow control in flexible manufacturing minimization of stockout cost, International Journal of Production Research 27 (1989) 701}715.
- [8] ISO/TC184 9506/1, Manufacturing Message Specification-Service definition, D.I.S., 1987.
- [9] ISO/TC184 9506/2, Manufacturing Message Specification-protocole specification, D.I.S., 1987.
- [10] Kenneth K. Boyer, Roger Hallowell, Aleda V. Roth, E-services: operating strategy—a case study and a method for analyzing operational benefits Journal of Operations Management, Research 20(2002)175-188.
- [11] Mac Carthy, Liu, A new classification scheme for flexible manufacturing systems, International Journal of Production Research 31 (1993) 299}309.
- [12] N. Mebarki, Une approche d'ordonnancement temps reHel baseHe sur la seHlection dynamique de re'gles de prioriteH, The'se de Doctorat, UniversiteH Claude Bernard Lyon I, January, 1995
- [13] P. Rolin, ReHseaux locaux, Normes et Protocoles, 5th Edition, Hermes, Paris, 1993.
- [14] Stecke, Solberg, Loading and control policies for flexible manufacturing systems, International Journal of Production Research 19 (5) (1981) 481}490.
- [15] Stecke, Kim, A flexible approach to part type selection inflexible flow systems using part mix ratios, International Journal of Production Research 29 (1) (1991) 53}75.
- [16] Tang, Yih, Liu, A framework for part type selection and scheduling in FMS environments, International Journal of Computer Integrated Manufacturing 8 (2) (1995)102}115.