Semi-heterarchical agile control architecture with intelligent product-driven scheduling

Theodor Borangiu*, Silviu Răileanu*, Damien Trentesaux, Thierry Berger****

* University Politehnica of Bucharest, Dept. of Automation and Industrial Informatics
ROMANIA (Tel: 40/214029314; e-mail: {borangiu, silviu}@cimr.pub.ro)
** Université Lille Nord de France, F-59000 Lille, UVHC, TEMPO Lab. F-59313 Valenciennes
FRANCE (Tel: 33/327511463, e-mail: {damien.trentesaux, thierry.berger}@univ-valenciennes.fr)

Abstract: The paper proposes a 2-layer holonic control architecture for optimizing production on both long- and short term. This architecture is intended for industrial environments affected by perturbations like: resource breakdown and recovery, exhausted part stocks, variable processing and transporting times. The proposed planning, scheduling and control models with their implementation frame are generic; the structuring of the decisional entities (Active Holon Entities – Expertise and Order) and the distributed decision making do not rely on specific or proprietary technologies. The method is based on: automatic switching between the global "batch" planning and scheduling horizon and the local "packet" scheduling horizon for resource (re)assignment – thus combining optimality, agility and fault-tolerance in business-oriented scenarios, and embedding holonic characteristics – autonomy and cooperation – in intelligent devices which assist products during their scheduling, execution and tracking – thus bringing closer the physical and decisional parts of entities composing a more performing, robust and agile control system.

Keywords: holonic manufacturing, decentralized scheduling, intelligent product, embedded devices

1. INTRODUCTION

Current advances in information technology and electronics made possible attaching various devices with decisional and communicational capabilities to almost all of the entities of a manufacturing system (FMS). This allowed moving from the classical centralized control approach to a fully decentralized approach where each entity (e.g. product, resource) has its own objectives, making it very hard for the system as a whole to achieve a global objective like minimizing the production time (makespan). Despite the structural differences between the two types of control architectures, some common aspects still exist, and only the way in which they are treated differs. These generic aspects are *production planning* and *resource allocation* (*scheduling*). To further optimize the production, a combination of the two problems has been proposed (Barták, 2000), but most of the classes of algorithms (operational research or artificial intelligence) offer good results only for a deterministic environment and a *runtime cost function*.

Since reality is rarely so deterministic, centralized approaches rapidly become inefficient when the target system must deal with a stochastic behaviour which may switch the primary objective of a designed system from global optimization to adaptability in the face of perturbations (Meyer, 2009) and real-time optimization. This led researchers to define other approaches to design control architectures that are self-organized, agile and fault-tolerant. This advance changed also the scheduling problem which is done now by several entities through coordination (Murillo et al., 2009) instead of being done centralized by a single entity. One of the new research directions in manufacturing control, that follows the previous guidelines, is based on the holonic manufacturing paradigm using autonomous and communicative information entities as resource counterparts, and on product-driven scheduling and execution control (Trentesaux, 2009).

Nowadays there exist several propositions based on holonic principles for structuring the elements of a manufacturing cell (Barata, 2006, Van Brussel et al., 1998, Raileanu, 2009), but there is a lack in the adaptation of these methodologies to the real world by suitable implementing frameworks. Some of them treat only isolated aspects of the theoretical models: (Borangiu et al., 2009) propose an implementing framework focused more on long term optimization with a centralized planning and scheduling entity that manages the decisional aspects of all the production orders; the research in (Raileanu et al., 2009) is focused more on real-time scheduling aspects (intelligent products negotiate with resources the allocation of operations). Since these two aspects tackled above do not interfere in the case of a real-life implementation (batches optimally planned and scheduled offline may undergo real-time changes), the paper proposes an original 2-layer semi-heterarchical architecture with distributed intelligence, able both to *globally optimize production planning and scheduling on long term* (at batch horizon), and to *locally reconfigure resource allocation on short term* for products in execution (at packet horizon) by real-time rescheduling **at shop floor level** through coordination between active holon entities.

The rest of the paper is structured as follows: section 2 presents the generic control model and the structure of the intelligent product (modelled as Active Holon Entity); section 3 gives a decentralized real-time scheduling method and the commutation criteria for planning and scheduling; section 4 describes an implementing frame using intelligent embedded devices; section 5 presents conclusions and future work.

2. THE SEMI-HETERARCHICAL CONTROL MODEL

2.1 The 2-layer planning and automation model

The target fabrication system consists of several autonomous workstations linked together by a conveyor. Each workstation contains one or more processing resources (CNC machines) and a part handling / processing resource (a robot) accessing the cell conveyor. The products are progressively processed and assembled in a number of workstations, being placed on pallets. Each pallet circulating on the conveyor is equipped with an Intelligent Embedded Device (IED) which is capable of memorizing information, communicating over an ad-hoc wireless network (Wi-Fi) with peer devices and taking real-time decisions regarding: *product scheduling* (allocating a resource to each operation), *product and resource tracking* (monitoring operation quality and resource performance, creating the product's "history"). IED also provides mobility of the control entity attached to the pallet on which intelligent products will automatically drive their manufacturing. The generic model for production planning, scheduling, execution control and traceability is organized on two layers (Fig.1):

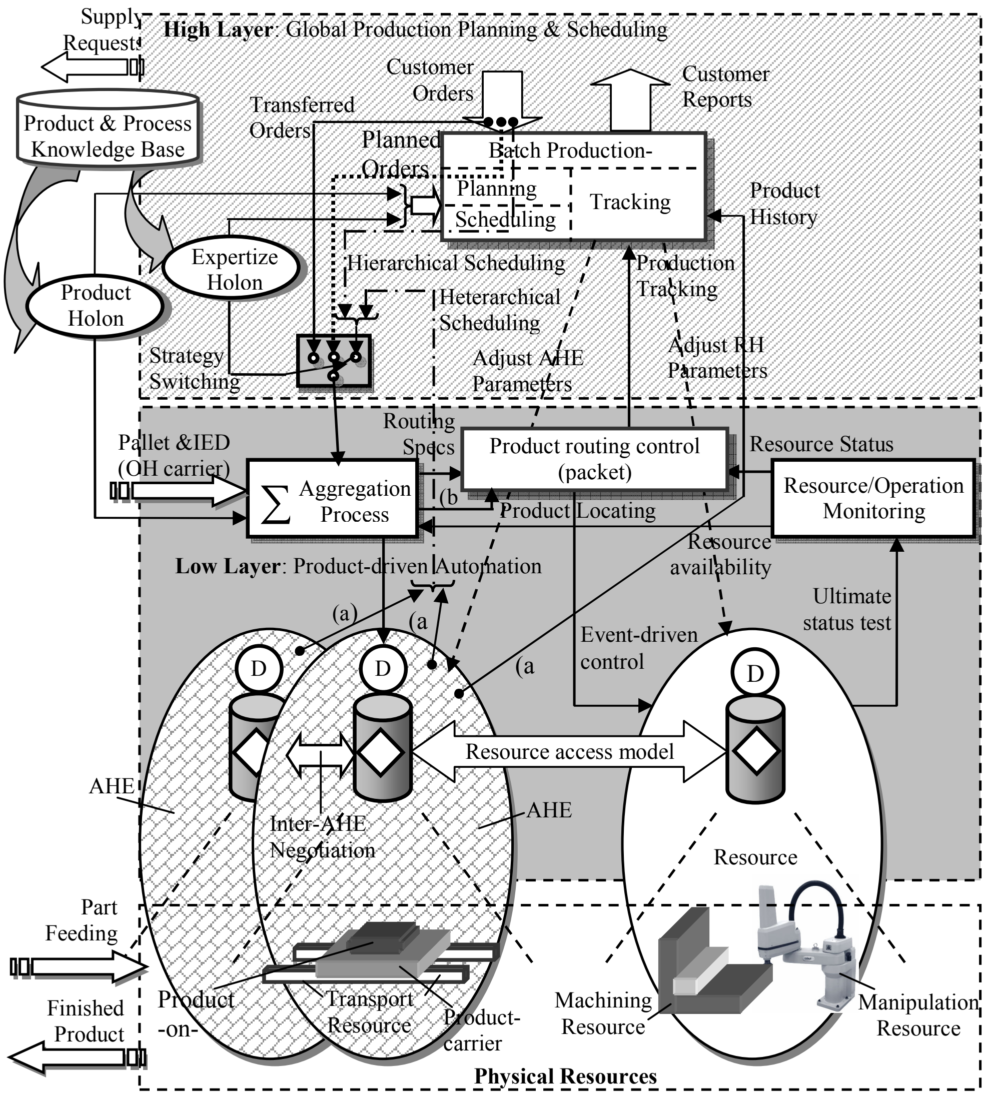

Fig. 1. The 2-layer generic model for production planning, scheduling, execution control and traceability

- A **high layer** in charge with collecting the clients' orders and performing the off-line decisional process of long term *planning and scheduling* them (at batch horizon). The layer is interfaced to the user by an interface for order reception and reports. The client's requests are mapped to an Aggregate Product Orders list (APO) from a Product and Process Knowledge Base which also generates the list of Product Holons – PH describing the services to obtain for the execution of each product type. The APO is input to a centralized application which, using Expertize Holons, generates the list of optimally ordered and scheduled (resource allocated for each operation) Order Holons (OH) – each one for an individual product. The optimization of the OH list is relative to a global cost function, at batch horizon, such as: makespan, resource loading, a.o.;

- A **low layer** in charge with *process automation* (OH execution), i.e. with *implementation of the production schedule* recommended by the high layer. This layer can switch its operational mode on request or automatically, following one Expertize Holon strategy, to *distributed decision for short-term scheduling* (at packet horizon of the products currently in execution) in order to react at unforeseen situations like: resource failure / recovery, bottlenecks / new available paths on the conveyor (due to last minute modifications in the schedule of certain OHs). Upon switching in this mode, **product-driven automation** is initiated; this means that *real-time heterarchical scheduling* (resource allocation) is done by the Intelligent Embedded Devices (IED) placed on the product carriers (pallets) and rend packet OHs active – i.e. Active Holon Entities (AHE).

This 2-layer model proposed for production planning, scheduling, control and traceability has a semi-heterarchical topology with **dynamic mode switching** between *batch optimality* in term of global costs and *packet cvasi-optimality* in terms of reactivity and adaptability to changes in the production cell. There are two types of components in the proposed holonic control architecture: (i) **active holon entity** (AHE) and Expertize Holon (EH) which take production decisions respectively based on multi-criteria negotiation (in real-time) and KB consulting (off-line); (ii) **passive holon entities** (Product Holon – PH and Resource Holon – RH) which are subordinated to the active entities.

The set of EH and the application of global batch production planning, scheduling and tracking act as a Coordinator Holon representing the high level control with all its attributes.

The Active Holon Entity (AHE) is an aggregate intelligent entity in charge of taking real-time decisions. It is composed of: (1) the product being fabricated, (2) the pallet which carries it and (3) an informational entity augmenting its decisional and communication capabilities. The associated informational entity is capable of taking decisions (D), which drive the production, and storing data (the diamond in Fig. 1).

A Resource Holon (RH) describes the physical resource (e.g. robot, conveyor), used for processing or transporting, along with its informational unit which is composed of a local D-part subordinated to the AHE which receives and interprets the operation granting decisions for program execution start.

The product- and process KB database stores the operations structure for the products the system is able to manufacture, in the form of PH - the "Services to Obtain" Holon (StOH).

A key role in this generic architecture is played by the AHE the structure of which comprises: a *module for memorization* of the product fabrication model (operations to be done, their parameters and precedence between these operations) and of the resource fabrication model (operations that can be done by each workstation and their times, the current status of the workstations and of the links between them); a *module for Wi-Fi communication* for inter-AHE negotiation (also used for product localization) and a *decision module* (for real-time mediator scheduling), see Fig.2. These three modules compose the Intelligent Embedded Device (IED) augmenting the OH to an active behaviour (Meyer, et al., 2009).

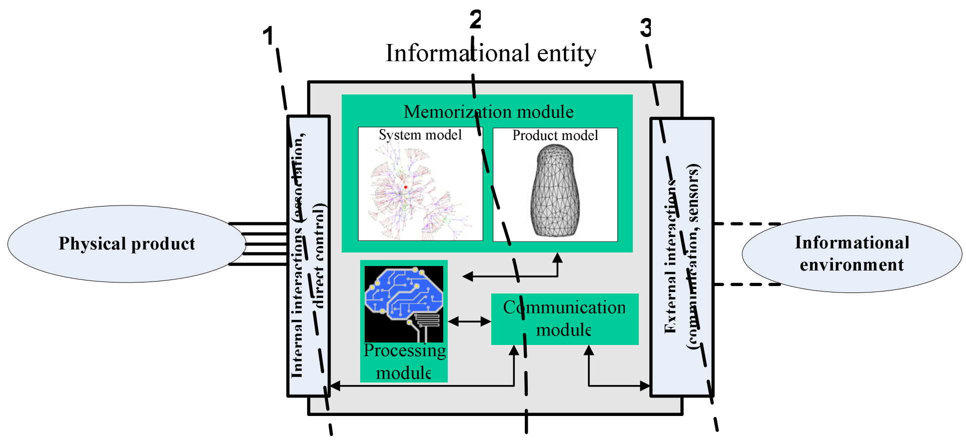

Fig. 2. Generic structure of the intelligent embedded device (IED) augmenting the OH with active behaviour

Two essential problems in real-time distributed, product-driven manufacturing control are *product localization* (line *b* in Fig. 1) and the *decision making for resource allocation* (lines *a* in Fig. 1). Both problems are influenced by the localization of the informational part of the entities playing a decisional role in real time scheduling with respect to the physical part (the product carrier) and by the synchronization solution between the two parts. In general, the augmenting entity can be structured in three ways, as suggested by the dotted-line separators 1, 2 and 3 in the generic IED structure represented in Fig.2:

1. *Intelligence located at distance*, on a remote server or on a group of servers for redundancy (dotted line 1). In this case the synchronization between the two parts is done through an *identification system* – usually RFID, which also serves for product localization since the sensors are located at well defined places (e.g. diverting points on the conveyor). Implementing technologies are available for this solution, but the control architecture tends to become centralized, a monolithic application being in charge of the decision making process, which has evident drawback and is not the scope of our design.

2. *Local intelligence* (dotted line 3), which makes the AHE entity more autonomous and the entire cell decentralized. The decision-making process is more agile since decision is taken near the point of interest, and more fault tolerant because in the case of a local failure the rest of the entities can continue to work. The entities do not rely in this case on the communication of control information but on the synchronization between them. Product localization is done in this case by the decision making module (IED) which interprets the signals received from sensors placed on the conveyor in the proximity of resources.
3. *Hybrid intelligence*: a remote main agent is in charge of taking important production decisions like planning and scheduling (e.g. the cell server), and local agents (IED or station computers) monitor and implement the decisions suggested by the main agent, resolving local problems and raising alarms in case they are not able to solve the problems. In this case product localization and decision can be done in any of the two above manners depending only on the existing infrastructure: if there is a global identification system, then the main agent can be put in charge of localization, and if there is a service oriented architecture, then the second approach can be used.

The second solution was chosen, since the hybrid holonic approach implemented in (Borangiu et al., 2009) showed that remotely placed intelligence consumed quite a significant time for rescheduling orders, mainly due to data transfer.

2.2 Manufacturing process

The aspect of manufacturing control this paper tackles is the fabrication process (**Area of interest** in Fig.3), and more precisely the optimal planning and agile scheduling of orders in the presence of disturbances. The manufacturing control process shows in Fig.3 the lifecycle of the client order:

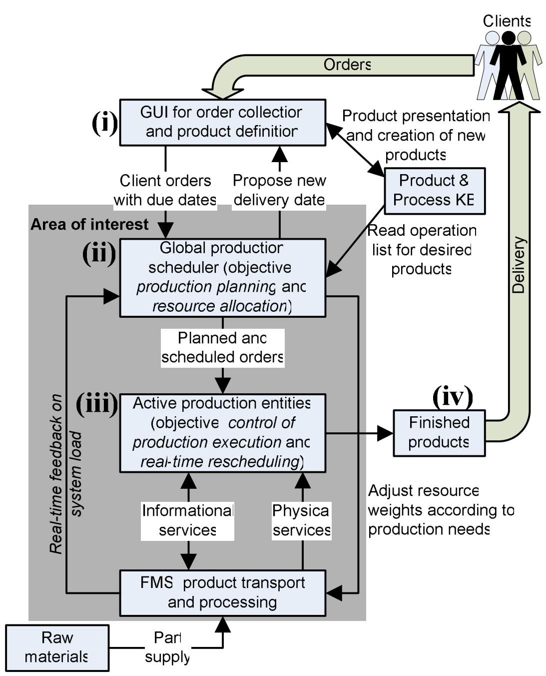

Fig. 3. The lifecycle of a client order

i. Production orders are *gathered* from the clients through a module that can serve also for *editing a product recipe.*
ii. Client orders are sent to the application in charge with batch production *planning and scheduling* (high level). This application computes how orders should be sent to the fabrication control system based on their due date and resources load. If a specific due date cannot be satisfied a new one is proposed to the client. Based on the "Earliest Deadline First" scheduling algorithm, *rush orders* are also considered for possible insertion among planned batches.
iii. Once orders are attached to the product carriers (through their associated IED), *execution starts*: upon an ultimate dialogue resource-OH the proposed schedule is followed until perturbations intervene; in this case a new schedule is computed in real-time by each AHE in coordination with other AHE in simultaneous execution (in the packet)
iv. Delivery of finished products to the clients.

3. REAL-TIME PACKET SCHEDULING

The low layer, in charge with short-term decision making, deals with the *real-time* aspects of the manufacturing system like: execution control, product- traceability and "history" reports, and decentralized scheduling (resource allocation in the absence of global scheduling or reallocation in case of disturbances). Rescheduling is done through coordination between the AHE, each entity that enters the process trying to optimize production from its own point of view. In this paper two major aspects are presented: the *commutation mechanism* between long-term and short-term scheduling and the *real-time cvasi-optimal scheduling mechanism* at packet level.

3.1 Commutation process

The IED units and the shop-floor PLC monitor the system's status; the events below trigger the commutation process:

i. If a resource breaks down and an AHE has operations allocated on it, it will need to reschedule these operations
ii. If there is a resource that can execute an operation faster than the current scheduled resource (which performs a task much later than expected), than the newly discovered workstation will be used. This decision is taken based on the current location of the AHE and on the system status model updated with the most recent information (resource states, intervals in which the resources are reserved by other products, transportation times)
iii. If there is a jamming on conveyor segment, the AHE must initiate a rescheduling process, trying to clear the transportation path – critical resource (the path to a resource from a certain point forward is usually unique)
iv. If a resource recovers from breakdown, both scheduling at packet level and for the rest of products should be done

3.2 Cvasi-optimal real-time resource allocation

Upon detecting one of the above events or when an explicit request is received, each AHE exits the existing scheduling and enters a real-time resource scheduling process according to one of the following two options (selectable):

1. <u>Searching the</u> nearest free <u>resource</u> <u>for the next operation</u>. This mode is the simplest since it considers only the operating time of the selected resource, its load and the pallet transportation time to it
2. <u>Scheduling all operations</u> when the product carrier enters the system (start of an AHE lifecycle), and the new AHE negotiates with the already executing AHE in the packet. A mediator agent is used for conflict resolution; all AHE perform a complete rescheduling of their not yet executed operations based on the current resource status model. The final real-time resource allocation is done by assembling the schedules proposed by each AHE (Fig. 4).

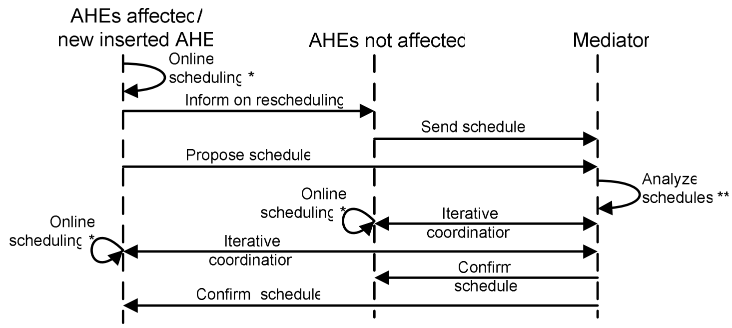

Fig. 4. Decentralized real-time resource allocation using a mediator agent

These individual schedules consider the constraints imposed by the transporting and processing resources: an AHE cannot surpass any other on the conveyor belt; only one AHE can be transported at a given time by a section of the conveyor and the resources can process only one AHE at a given time.

Upon detection of a rescheduling event the AHE send their resource allocation (*online computed* or *default* if not affected by the event) to a designated agent – the mediator – who must globally analyze the resources allocation and eliminate conflicts. This process takes into account the demands of each participating AHE and uses dialogue to reach a common agreement. During online allocation AHE will wait for the mediator's confirmation after which production will resume. The AHE that currently execute an operation (they receive a service at a workstation) will propose a schedule starting from the next operation and continue on with their execution.

(*), **AHE on line scheduling algorithm**:

1. Invalidate schedule
2. default_operation_and_resource:=nothing;
3. *For each possible processing operation* of the AHE not scheduled
   - 3.1. *For each resource* able do the selected operation
     - 3.1.1. If there are constraints from the moderator following an iterative coordination then *delay the time of current selection* (scheduled time range) by the requested time
     - 3.1.2. If the time of current selection is less then the time of default selection then *choose operation and resource*:=(operation selected at 3, resource selected at 3.1)
4. *Go to* 2 *and repeat* until all operations are scheduled

(**), **Analyze schedules** (AHE-mediator conflict resolution)

1. Form the resources GANTT chart using the proposed resource allocation from each AHE;
2. For each resource validate the proposed allocation starting with the AHE that arrives first; invalidate the proposed allocations that overlap and start a coordination dialogue with the associated AHEs

The mediator, a key element in decentralized scheduling, is elected *dynamically* from the existing AEHs at the beginning of the production control and each time the current mediator is no more available. This process uses a simple algorithm: the mediator is elected based on the time of insertion of each AHE in the production system, the oldest one in the system being elected the mediator.

4. IMPLEMENTATION FRAMEWORK

The implementation of the proposed type of scheduling was done on the pilot platform in the Robotics and AI Laboratory of University Politehnica of Bucharest as a second research stage to further develop real-time production control relative to the RVHOLON project (www.rvholon.cimr.pub.ro) and was funded by the National Council for Scientific University Research, in the framework of the National Plan for Research, Development and Innovation, grant 69/2007.
According to the 2-layer control model described in Fig.1, packet scheduling was decoupled from long-term planning and scheduling. This decoupling is supported by aggregating a *product* with a *pallet carrier* equipped with an *IED*. The physical implementation of the AHE is shown in Fig. 5.

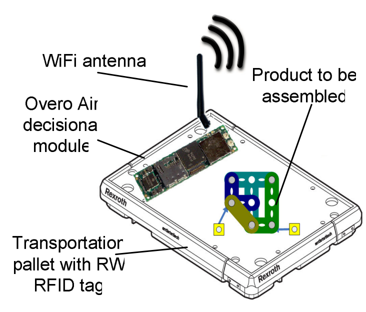

Fig.5. Physical implementation of an AHE

Hence, an AHE is composed from:

1. An *Overo Air decisional module* (http://www.gumstix.net) running Linux configured for real-time applications and for power consumption optimization because it runs on battery power; it is provided with WiFi communication

capabilities allowing each AHE to be in permanent contact with other devices: other AHE for real time scheduling, global cell server from which optimal recommended schedules are received and to which reports about product history and production status are sent, and RH for adjusting their parameters, receiving their status model and sending fabrication and transportation commands (Fig.6)

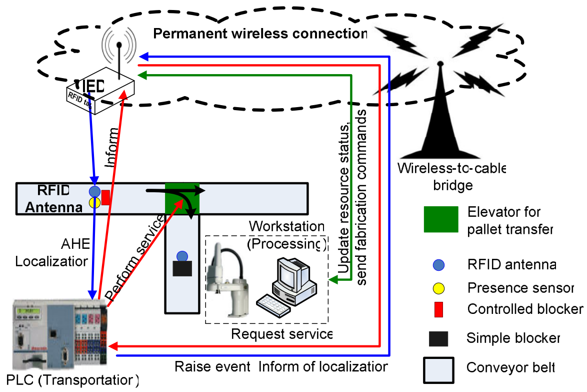

Fig. 6. Product localization and scheduling with AHE

2. *WiFi antenna* which is part of the Overo Air IED kit and extends the area of the wireless communication
3. *Transportation pallet with RW RFID tag*: this entity is the carrier of the product to be progressively manufactured, offering it transportation services
4. *Product*: the part of the AHE being assembled and at the en provided to the client.

After online scheduling done by the AHE, two processes are important when AHE obtain services – routing control of the attached products and processing operations upon them.

The infrastructure supporting the high level control consists of the PCs attached to the workstations and the cell server on which the global planning and scheduling application resides; this application can be relocated on any PC connected to the cell network infrastructure. The *product routing control* (low level) is done by a PLC which receives from AHE standard files the content of which is used to command the conveyor devices (motors, diverting units, stoppers) so that the product visits the allocated resources and gets processing services.

Product localization is done by the PLC which reads the IDs of the pallets in fixed (e.g. diverting) places using the RFID system (*AHE Localization*), and offers this information to the exterior through a server. This information is then read by the AHEs which are continuously polling the PLC; when their own ID is detected by the PLC the location where the ID was read is associated with the corresponding pallet (*Raise event: Inform of localization* in Fig. 6).

The localization event triggers a decisional process on the AHE which sends its decision to the PLC (*Request service*), this entity being in charge of its realization (*Perform service*). After completion of the requested operation, the PLC informs the AHE on the result (*Inform*) and why the result is negative.

5. EXPERIMENTAL RESULTS

The proposed 2-layer control model was tested in the pilot FMS platform of the Robotics & AI lab of the University Politehnica of Bucharest. The cell is composed of 4 material processing stations (robots, CNC), a pallet supply station and a part supply station linked by a cell conveyor. Experiments were carried out; as an example, Fig. 7 shows a batch of 8 products of 4 distinct types which were planned, scheduled and executed in packets of 4. A resource breakdown was simulated during execution, (flash in Fig.7) causing resource rescheduling (see the two GANTT product charts).

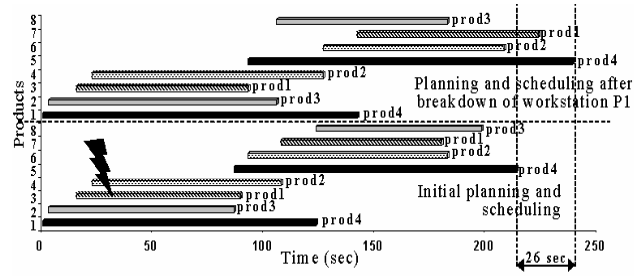

Fig.7. Execution times before & after a resource breakdown

The complete batch execution shows that the control system performs well even if affected by perturbations (only 26 sec, i.e. 10% increase of the total time, no interruption). Product prod3 on pallet 8 is executed faster after on line rescheduling, but makespan takes longer because only 3 of 4 resources are available. Product rescheduling switches automatically to the on line mode, triggered by 2 event types: station breakdown, and missing parts. Experiments carried out on a batch production of 256 products put in evidence recovery times of 6.4 to 6.8 time units from resource failure to rescheduling of packet OH in execution [packet=5] and resuming production, and of 83 to 136 time units from local storage depletion to generation of a Supply Holon, routing it to the empty storage, automatic refill by the station robot and resuming production.

REFERENCES

Barata, J., The Cobasa architecture as an answer to shop floor agility, manufacturing the future - concepts, technology, *Visions. Pro literatur Verlag*, 31-76, 2006

Barták, R., Mixing planning and scheduling to model process environments, *PACLP*, Manchester, UK, 2000

Borangiu, Th., Gilbert, P., Ivanescu, N.A., An implementing framework for holonic manufacturing control with multiple robot-vision stations, *Engineering Applications of AI* (EAAI), **22**, *4-5*, 505-521,Elsevier, 2009

Brussel H.V., et al., Reference architecture for holonic manufacturing systems: PROSA, *Computers in Industry*, **37**, *3*, 255–274, 1998

McFarlane D., Sarma S, Chirn J.L., Wong C.Y., Ashton K., The intelligent product in manufacturing control, *Journal of EAIA*, July 2002

Meyer, G.G. et al., Intelligent Products: A survey, *Computers in Industry*, doi:10.1016/j.compind, Elsevier, 2009

Murillo J., Schedule coordination through egalitarian recurrent multi-unit combinatorial auctions, *Applied Intelligence*, doi:10.1007/s10489-009-0178-7, 2009

Raileanu, S. at al., The open-control concept in a HMS, *Proceedings of RAAD'09*, Brasov, 2009

Trentesaux, D., Distributed control of production systems, *EAAI*, **22**, *7*, Elsevier, 971-978, 2009