7^e *Conférence Internationale de MOdélisation et SIMulation -MOSIM'08 - du 31 mars au 2 avril 2008 – Paris – France*
Modélisation, Optimisation et Simulation des Systèmes : Communication, coordination et coopération

MODELLING AND DESIGN OF A HOLONIC MANUFACTURING CONTROL SYSTEM WITH OPTIMAL JOB SHOP SCHEDULING

Theodor Borangiu, Silviu Raileanu, Nick Ivanescu

University Politehnica of Bucharest, 313, Spl. Independentei, Bucharest (e-mail: borangiu@cimr.pub.ro)

ABSTRACT: *The paper proposes the model of a job shop type production system containing multiple networked robot workstations operating under visual control, for which a heuristic algorithm is used to schedule the operations set at batch manufacturing level. It is assumed that transportation times of products between workstations cannot be neglected. A holonic manufacturing control architecture is designed, based on four types of entities: Resource Holons, Product Holons, Order Holons and an Expertise Holon containing: a Global Scheduler, System Monitoring and Database which generate optimal production plans. These entities are embedded in a computing and control structure, being interconnected by a 4-layer fault-tolerant communication network keeping track of job execution and resource status. Implementing and experimental results are reported.*

KEYWORDS: *job-shop modelling, job scheduling, holonic manufacturing control, part transportation, robot-vision*

1. INTRODUCTION

To be competitive, manufacturing should adapt to changing conditions imposed by the market. The greater variety of products, the possible large fluctuations in demand, the shorter lifecycle of products expressed by a higher dynamics of new products, and the increased customer expectations in terms of quality and delivery time are challenges that manufacturing companies have to deal with to remain competitive. Besides these market-based challenges, manufacturing firms also need constantly to be flexible and adapt to newly developed processes and technologies and to rapidly changing environmental protection regulations [Anton, 2007].

A proposed solution to this problem is a distributed architecture, in which information entities, having manufacturing counterparts, cooperate to solve together the assigned tasks. This type of control architecture is called holonic manufacturing system, being developed in the frame of the RVHOLON (Robot Vision Holonic Manufacturing Control) grant [www.rvholon.cimr.pub]. The term "holon" was first proposed by Arthur Koestler and it is a combination of the Greek term *holos* (whole) with the suffix on which suggests a particle. Based on Koestler's concept, the following definitions were established by the Holonic Manufacturing Systems (HMS) consortium [Van Brussel, 1998]:

- *Holon*: An autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. It consists of an information processing part and a physical processing part. A holon can be part of another holon.
- *Autonomy*: The capability of an entity to create and control the execution of its own plans and strategies.
- *Co-operation*: A process whereby a set of entities develops and executes mutually acceptable plans.
- *Holarchy*: A system of holons that can co-operate to achieve a goal or objective. The holarchy defines the basic rules for co-operation of holons and thereby limits their autonomy Cheng, 2006; Wyns, 2007].
- *Holonic manufacturing system (HMS)*: a holarchy integrating the entire range of manufacturing tasks from order booking to design, production and marketing to do the agile manufacturing enterprise.
- *Holonic attributes*: attributes of an entity that make it a holon. The minimum set is thus autonomy and cooperativeness [Bongaerts, 1998, Morel, 2003].

The following objectives of the holonic manufacturing control were proposed in the RVHOLON project:

- stability in face of disturbances (resource failures);
- adaptability and quick response in face of changes (client orders);
- in line product quality control by artificial vision;
- efficient use of available resources (robot, CNC).

The objectives of this paper are: the definition and set up of the holon structures, the design and implementing of an operation scheduling mechanism in a holonic, fault-tolerant manufacturing control system networking multiple robots visually guided and CNC machines for material conditioning (processing, assembling, testing). The control provides on-line adaptation of robots to the work environment, material flow and task specifications by applying OO and AI techniques to machine vision of the robot controllers. The job scheduling algorithm gives the best sequences for the currently available resources.
The paper is structured as follows: in section 2 the proposed architecture of the system is described; section 3 presents the composition and data structures of the holons, section 4 describes the job scheduling algorithm; in section 5 the holonic mechanism for job rescheduling at change detection is proposed; finally, section 6 includes implementing issues, the solution for fault

tolerant control, experimental results obtained by simulation, conclusions and perspectives of future work.

2. SYSTEM ARCHITECTURE

The holonic material conditioning structure is composed by a number or robotic resources, interconnected by a closed-loop transportation system (conveyor). The final product results by executing a number of mounting, joining and fixing operations by one or several of the networked robots. The operations are scheduled off-line or rescheduled online by a holonic bidding mechanism proposed in Section 5. The set of specific production operations is extended to on-line part conditioning (locating, tracking, qualifying) and checking of relative positioning of components and geometry features. These functional extensions may be supported by artificial vision either integrated with motion control (Guiding Vision for Robots - GVR) or as stand alone, computer vision (Automated Visual Inspection – AVI). In both cases, vision is used online to check for proper geometry features and presentation of assembly components in view of robotized mounting, and for inspecting the product in its intermediate and final execution stage – positioning, alignment, completeness [Borangiu, 2004].

Traditional networked conditioning structures have either a hybrid or heterarchical architecture. The first one, derived from the hierarchical architecture, allows cooperation and sharing of information between lower-level (robot) controllers; a supervisor initiates all the activities and then the subordinates cooperate to perform them. The second is formed by a group of independent entities called agents that bid for orders based on their status and future workload. There is no master-slave relationship; all the agents including the manager of a particular order are bidding for it. Due to the decentralized architecture, the agents have full local autonomy and the system can react promptly to any change made to the system. However, because the behaviour of an order depends on the number and characteristics of other orders, it is impossible to seek global batch optimization and the system's performance is unpredictable [Deen, 2003; Woolridge, 2002].

In order to face resource break-downs, the networked production structures discussed here uses robot controllers with multiple-network communication facilities allowing for fault-tolerance: targeted data saving and task redistribution. This is why we propose an holonic architecture in which holons group together and form holarchies depending on the status of the system: if each entity works according to the initial plan (off-line computed by the scheduler), than the system behaves in a hierarchical way; if something goes wrong or a change occurs, the entities regroup, the plan is rescheduled, and the system behaves heterarchically. These two behaviours can be seen as two holarchies.

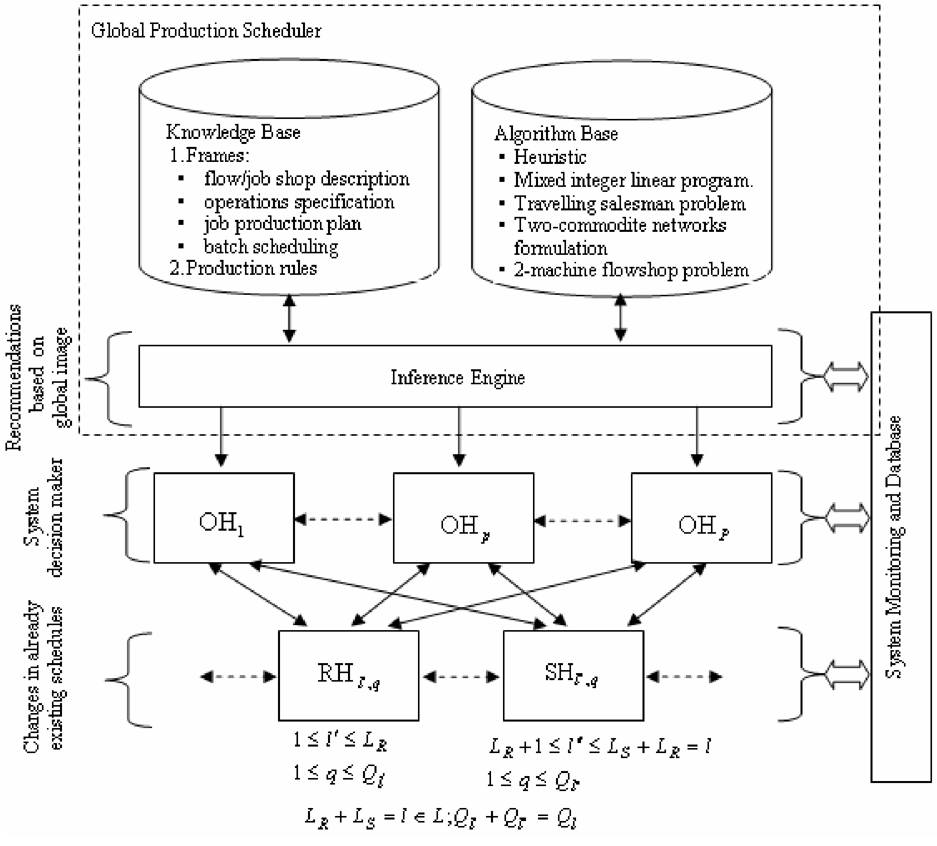

Figure 1. Knowledge-based holonic architecture for job shop robotized production.

The flow of information between control units is always bi-directional due to the decentralized nature of the architecture and the application of the holonic concept of cooperation. This production control structure is derived from a generic manufacturing one, formed by four types of entities, as shown in Fig. 1 [Babiceanu, *et al.*, 2004; Borangiu, 2007]:

1. A layer of *Order Holons* $(\text{OH}_p, 1 \le p \le P)$ of variable depth, corresponding to production plans off-line computed for the P final products.
2. Two types of Resource Holons:
   - *Robot* (or Material Conditioning) *Holons* $(\text{RH}_{l',q})$, formed by all robot manipulators, grippers and tools together with their controllers, responsible for mounting, fixing and fastening assembling components, and for moving their arm-mounted cameras in picture-taking points where the products are visually inspected [Borangiu, 2004].
   - *Sensory* (Material Tracking & Checking) *Holons* $(\text{SH}_{l'',q})$, formed by all machine vision systems and magnetic code RD/WR devices respectively used for component / subassembly position and geometry control and product tracking on pallets.
3. A *System Monitoring and Database* (SMDB) entity, responsible for monitoring the number of jobs and availability of resources in the system, and for keeping track of the already executed operations and assembly jobs.
4. A *Global Production Scheduler*, generating production plans for all batch products in the form of OH. An algorithm base is embedded into a knowledge-based system (KBS). An inference engine in the KBS controls, according to a forward chaining control strategy, the procedures of: triggering production rules, switching to algorithmic schedule generation, and holonic fault-tolerant task allocation [Bongaerts, 1996].

3. SEMIHETERARCHIC MANUFACTURING ARCHITECTURE AND HOLON STRUCTURE

The holonic control and command solution proposed in this project provides a set of functionalities rending the material-conditioning cell flexible, rapidly reacting to changes in client's orders (batch size, type of products, alternate technologies, updated programs), and fault-tolerant to resources getting down temporarily. In fact, the holonic control architecture proposed follows the key features of the PROSA reference architecture [Van Brussel, 1998; Valkaenarts, 1994], extended with:

- real-time, in line quality control of products in different stages of manufacturing;
- follow-up and qualifying of materials and subassemblies circulating on pallets within the shop-floor manufacturing cell;
- robotized material processing (e.g. assembling, grinding, cutting) under visual control/guidance.

Figure 2 shows the data and functions included in the three basic types of holons:

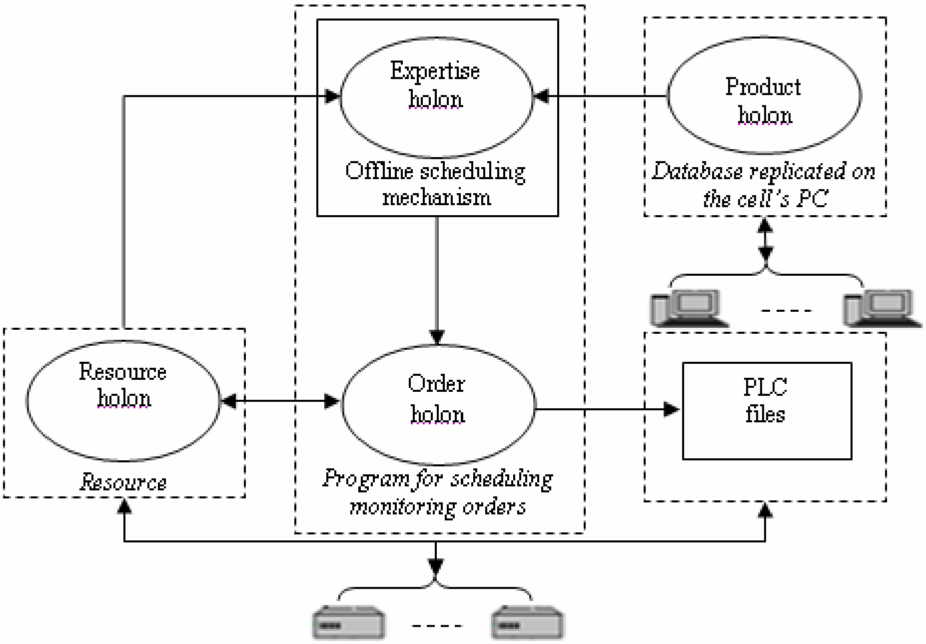

Figure 2. Relevant holons in the job shop-type manufacturing system.
Interactions between holons and other system resources and the physical allocation

There are three basic types of holons aimed to provide production control: *resource holons*, *product holons* and *order holons*. To these basic holon types, a fourth one is added: the *expertise* (decision) *holon*, which is responsible to calculate off-line and provide optimal production plans at batch level [Barata, 2005].

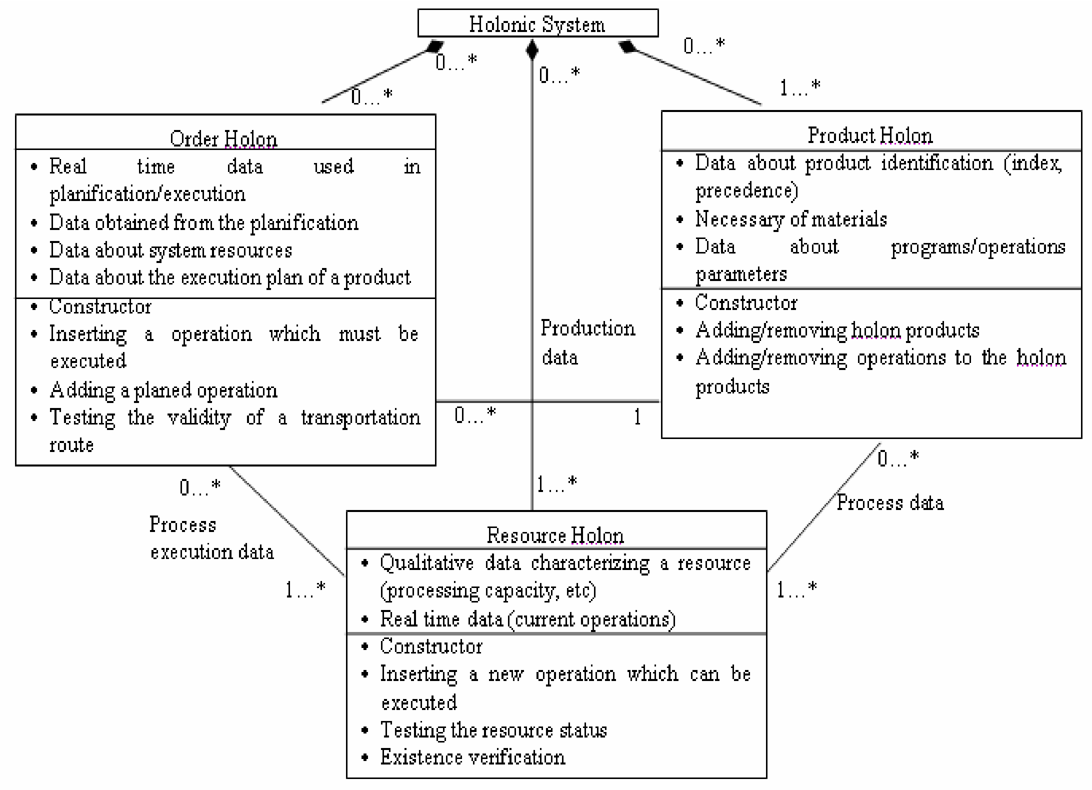

Figure 3. Data and functions embedded in the basic types of holons: product, resource and order.

Production modelling is done starting from the operations which are executable in the system; next, these operations are associated to (robot-vision) workstations, and finally one creates products from these defined associations (Fig. 2). Because operation and resource indexes are used, indexes must conform to a standard structure and be at the same time globally consistent and valid; thus, operations will be indexed in increasing order, as well as resources added in time.

The data structures representing the three basic holons contain the following information:

- *Order holons*: characterized by information of the following types: identifier, production information (operations to be performed upon the current holon, the production plan being embedded in the order holon), planning support information (these are variables indicating the state of a product at different time moments), information resulting from the Global Scheduler (vectors describing the successions of operations, precedence, necessary resources carrying out the operations and processing times).
- *Resource holons* contain the following information: identifier, operations that the attached resource is capable to do, state of the resource, current operation and the holon it belongs to (a resource can process a single operation at a given moment of time), Table 1.
- The *product holons* consist of a database storing: the identifier of the product, a list of operations to be executed on the product, the precedence of the operations to be executed, the necessary materials and resources, the programs and the parameters needed to execute the operations [Duffie, 1994].

4. JOB SCHEDULING ALGORITHM

When trying to optimize a schedule ones objective is to minimize or maximize one of the following objective functions [Iwamura, 2005]:

- *Efficiency of a machine* (how intense is the specified resource used – machining time/total processing time) [Onori, 2006; Leitao, 2007].
- *Production time* (total amount of time needed to process the entire production).
- *Cost of machine* operating (machining cost of resources) [Tanaya, 1995; Leitao, 2006].

The algorithm presented here can be regarded as a multi criteria scheduling algorithm because on one hand it tries to minimize the production time function and on the other hand it finds the appropriate transporting operations that must be performed (minimize the waiting time).

4.1. The system model

The requests for the scheduling problem are received as processing operations that should be executed to obtain the final product and the scheduling algorithm has to construct the optimum path of the pallet throughout the system consisting of processing and transport operations. This schedule takes into account the occupation and future workload of the workstations and generates a

Table 1. Class Resource Holon

```
Resource holon
//Resource characteristics
- string name
- int index  //A resource can be identified by name or by index
- int operation_number
- public string[] operations = new string[maximum_operations]
- public int[] operation_type = new int[maximum_operations]
- public float[] operation_time = new float[maximum_operations]

//Informations needed for planification
- int reserved_for  //The index of the order holon for which the resource is reserved
- string current_opperation
- int current_holon
- int current_state
//Methods for “Resource holon” class:
+ public resource_holon()  //Constructor
+ public resource_holon(resource_holon rh)  //Copy constructor
+ public void add_operation(string op_name,int op_time)
+ public bool test_resource()
+ public bool test_operation(string op_name)
```

sequence of operations, with precedence between them, which is in fact the production plan of a single product.

The algorithm uses a lookup table in which the minimum paths between the workstations are stored; these paths are stored as operations that must be executed. Physically, a path is a section of the conveyor which can perform transportation operations, but only one at a given time. Internally the HMS is seen as an oriented graph in which the resources are the nodes of the graph and the connections between the resources are represented by the arcs. Thus, the algorithm behaves well, and adding or removing workstations or sections of the conveyor will be easier hardware and also software.

During the planning process the algorithm models each type of holon as a class with the following proprieties and the connections between the resources are represented by the arcs. In this way the algorithm can scale well, and adding or removing workstations or conveyor sections will be done easier [Ramos, 1996].

The minimum path is used by the algorithm for choosing the fastest (lowest cost) way of transporting the palette between workstations throughout the system. The operations are chosen according to the processing time, at which is added the transportation time. This information is given by the lookup table, in order to obtain the whole time these two times have to be added.

4.2. The pseudo code of the scheduling algorithm

A basic (quasi optimal) production plan is generated and embedded in the Order Holons (OH) for each final product. The heuristic Global Scheduler uses the notations and definitions below:

- Set of unscheduled Order Holons (M_{np})
- Set of scheduled Order Holons (M_p)
- Set of Order Holons chosen for scheduling (M_a)
- Set of Resource Holons (M_r)
- Set of operations belonging to the order holon with the index *ind* waiting to be scheduled (M_{ind,to_plan})
- Set of operations belonging to the order holon with the index *ind* that had been scheduled ($M_{ind,plan}$)

A production plan of a product is computed offline by the Global Production Scheduler and embedded in the resulting Order Holons (OH) as an ordered vector of triplets, each specifying the operation to be processed, the resource on which it is processed and the processing time of the operation [Morel, 2003; Lastra, 2006].

One operation $op_i \in M_{ind,to\_plan}$ in the ind^{th} OH is *schedulable* at time *t* if:

1. No other operation (mounting, inspecting) upon the same product is being processed at time *t*.
2. All operations preceding op_i have been completed before time *t*.
3. At least one resource needed to carry out this operation is defined as *operational*.

Additionally, if more than one operation can be processed at time *t*, than it will be chosen as the first operation with the minimum processing time.

Both the off-line Global Scheduler and the real time holonic control mechanism need to know the current operation status (os_i), retrieved respectively from input planning data and from diagnosis/sensory data:

$$os_i = \begin{cases} 0, & op_i \text{ is scheduled} \\ 1, & op_i \text{ is chosen for scheduling} \\ 2, & op_i \text{ has not been schedualed} \end{cases}$$

and also the resource status (rs_i)

$$rs_i = \begin{cases} 0, & \text{resource } i \text{ suffered a breakdown} \\ 1, & \text{resource } i \text{ is operational but unavailable} \\ & \text{because it is occupied by another order} \\ & \text{holon} \\ 2, & \text{resource } i \text{ is operational and available for} \\ & \text{work} \end{cases}$$

The heuristic scheduling algorithm for N products, not necessarily identical and each containing a number of n_i ($i = 1,\ldots, N$) operations executed on R resources is given below. The algorithm will be run for tuning several times with a different maximum number of pallets accepted simultaneously in the shop floor system (Fig. 4).

```
1. for $i$=1 to 11 ; theoretically, the maximum number of pallets that may coexist in the system is 9
    1.1. - time=0
         - $M_{np}$ = all the order holons that should be scheduled
         - $M_p$ = {empty}
         - $M_a$ = {empty}
         - $M_r$ = all the resource holons that should be considered when scheduling
    1.2. As long as (cardinal ($M_{np}$) > 0 or cardinal ($M_a$) > 0)
        1.2.1. Is the number of order holons that are in the planning stage greater or equal than $i$ (the maximum number of pallets that can coexist in the system) ?
            1.2.1.1. YES – Chose an operation with the minimum execution time from $M_a^*$
            1.2.1.2. NO – Chose an operation with the minimum execution time from $M_a \cup M_{np}^*$
*When choosing an operation, the execution time is calculated taking also into consideration the transportation time on the conveyor to that resource.
        1.2.2. Has the processing operation chosen been selected before ?
            1.2.2.1. NO – Insert into $M_{ind,to\_plan}$ the transport operations that will bring the $ind^{th}$ order holon to the processing resource selected in step 1.2.1. along with the precedence of these operations. Select as current operation the transport operation that leads to the resource
            1.2.2.2. YES – go to 1.2.3
        1.2.3. Add the selected order holon into $M_a$, if it is not there yet, along with the selected operation, resource and time for processing and set the operation status to chosen for execution
        1.2.4. If the operation belongs to an order holon from $M_{np}$ eliminate that holon from $M_{np}$
        1.2.5. Set the state of the precedent resource to operational and available and the state of the current resource to operational and unavailable.
        1.2.6. If any other operations can be selected for scheduling / execution, than go to Step 1.2.1., otherwise continue to Step 1.2.7.
        1.2.7. Calculate the minimum completion time ($t_{min}$) of the operations chosen for scheduling and subtract it from the completion time of the other operations chosen for scheduling. The operation with $t_{min}$ will advance in the state completed
        1.2.8. Update the time of the system (time = time + $t_{min}$)
        1.2.9. If there are order holons for which all the processing operations have been completed, than add the necessary transport operations for evacuating the corresponding holons from the system
        1.2.10. Are there holons that have been evacuated from the system ?
            1.2.10.1. YES – add them to $M_p$ and delete them from $M_a$
        1.2.11. STOP (from 1.2)
    1.3. STOP (from 1.)
```

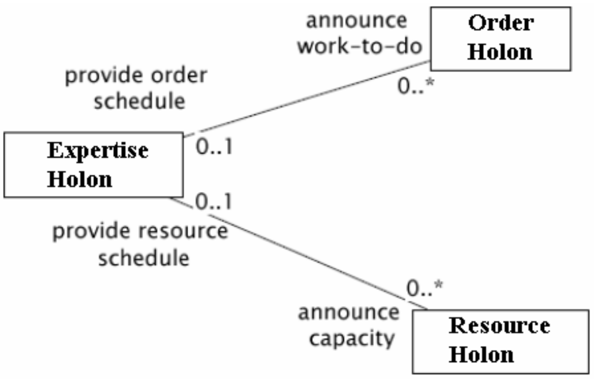

Figure 4. Interaction between the algorithm (expertise holon) and the order and resource holons.

5. REAL TIME PRODUCTION UPDATE UNDER HOLONIC MECHANISM

The off-line scheduling (execution of expertise holons) can be expanded to real-time, on line scheduling because the runtime of the algorithm is relatively small. Once the production schedule is computed, this information will be transferred to the execution mechanism – the PLC controlling the transportation system – as a binary file which contains holons, operations and resources indexes.

In the event that the production plan is invalidated by changes in the system (e.g. a resource is not available),

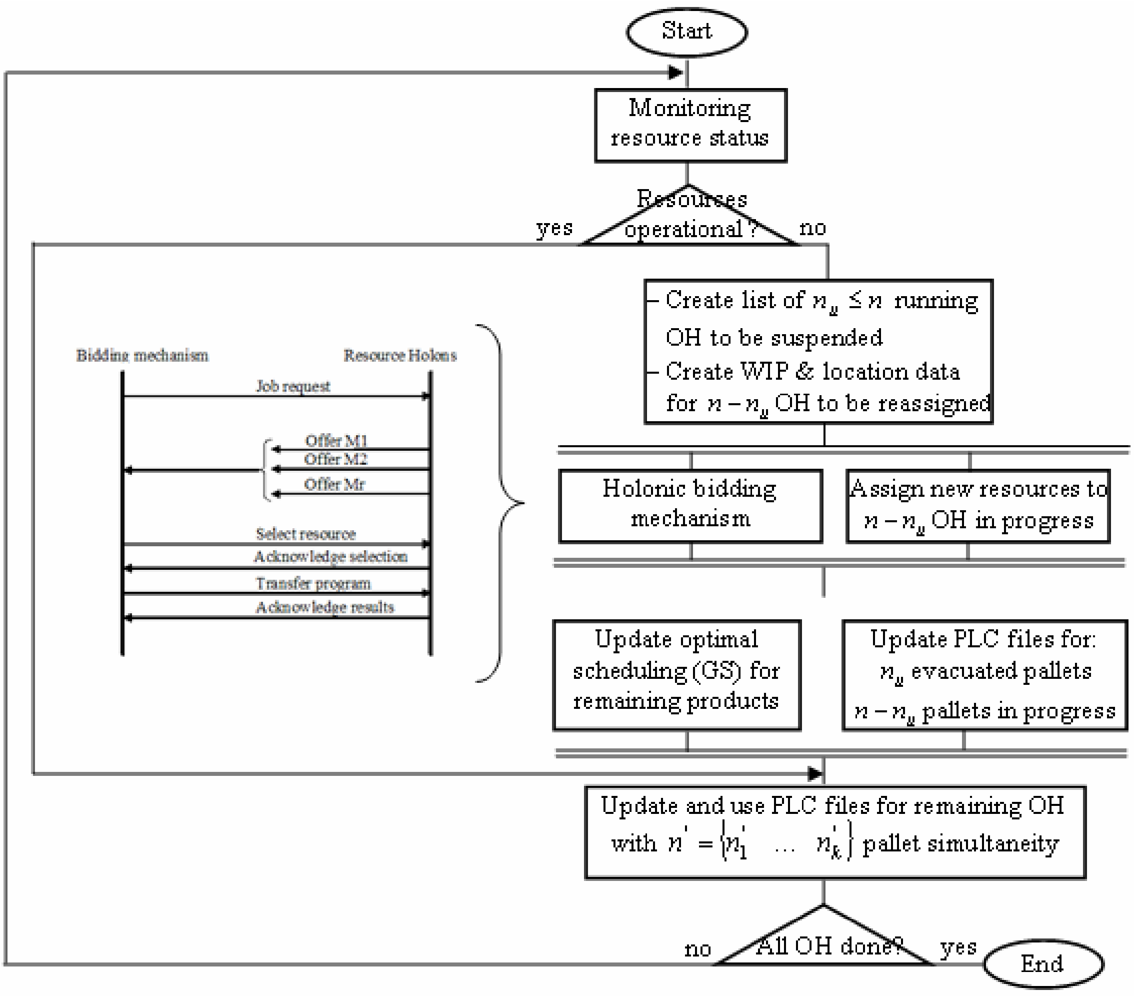

Figure 5. The real time holonic production control mechanism

which affect the offline scheduling, then the control switches to an online scheduling (Fig. 5) in which the remaining undone operations from the order holons are reallocated to the valid resources according to an auction process at which will participate both the entity administrating the order holons and the resource holons which are valid and running [Pétin, 2007].

No new pallets are introduced in the system until all the operations not yet done for the products already in execution are allocated and executed. Simultaneously with this auction a new offline scheduling which takes into account the new working capabilities of the system is calculated for the remaining products in the batch. The new offline scheduling will be used after the completion or evacuation of all the pallets that were in the system when the resource breakdown took place. After a resource breakdown, some types of products may no more be executed. In this case the corresponding pallets will be evacuated with an *unfinished* status and the rest of the order holons waiting to be inserted into the system and being of the same product type will be not taken into account by the expertise holons (when rescheduling).

The program that administrates the bidding mechanism should have a facility for monitoring the state of the resources in the system. In the event that a resource breaks down or a signal which invalidates the offline scheduling is received, than the online bidding mechanism takes the place of the offline scheduling. This monitoring program should run on a machine which must be continuously connected to the system processing resources – the cell server (replicated on another machine to provide fault tolerance in case of crush).

For each product to be executed there is a dedicated order holon with specific responsibilities, including the right to select resources which will execute the necessary operations. This decision is made by the manager of the new job – the associated Order Holon (OH) – based on the proposals received from the Resource Holons (RH) and Global Scheduler (Expertise Holon – EH).

Once created the OH layers for each individual product, a holonic decision-making process for incoming jobs operates in real production time. The entities involved in this process for any incoming job are the OHs, RHs, and the real-time holonic mechanism [Kusiak, 1990]. In the holonic mode, the production and control operations are executed by the RHs after an algorithmic evaluation of the requests received from the OHs by the distributed program that administrates the bidding mechanism.

6. CONCLUSION. EXPERIMENTAL RESULTS

The holonic control mechanism in tandem with the knowledge-based global scheduler is currently under implementing in a manufacturing cell with five networked robot-vision stations, as shown in Fig. 6. The five robotic stations are interconnected by a Bosch Rexroth TSplus closed-loop twin-track, pallet-based power-and-free conveyor with four linear bidirectional derivations that move subassemblies fixed on pallets in single or double access production stations: W1, W2, W3 and W4 run by the industrial robots.

In order to verify and tune the job scheduling algorithm presented in section 4, the following distribution of operations was defined:

- Station 1 (W1): operations 1 and 2;
- Station 2 (W2): operations 1 and 2;
- Station 3 (W3): operations 2, 3 and 4;
- Station 4 (W4): operations 1, 3 and 4.

A number of four types of products must be executed; their operations belong to the operation set {1, 2, 3, 4}:

- *Product 1* with the operations set: operation 1 without precedence, operation 2 without precedence and operation 3 must be executed after operation 1.
- *Product 2* with the operations set: operation 1 without precedence, operation 2 must be executed after operation 1, operation 3 must be executed after operation 2, and operation 4 must be executed after operation 3.
- *Product 3* with the operations set: operation 1 without precedence, operation 2 must be executed after operation 1, and operation 3 which must be executed after operation 1.
- *Product 4* with the operations set: operation 4 without precedence, operation 1 must be executed after operation 4, operation 2 must be executed after operation 1, and operation 3 must be executed after operation 2.

For the defined production structure and the 4-product batch, job scheduling was run for a different numbers of pallets coexisting in the production cell, simultaneously travelling on the conveyor. The ensemble [Robot Work-stations (W) – Conveyor branches (C)] in Fig. 6 was modelled as an oriented graph with R-nodes (Fig. 7).

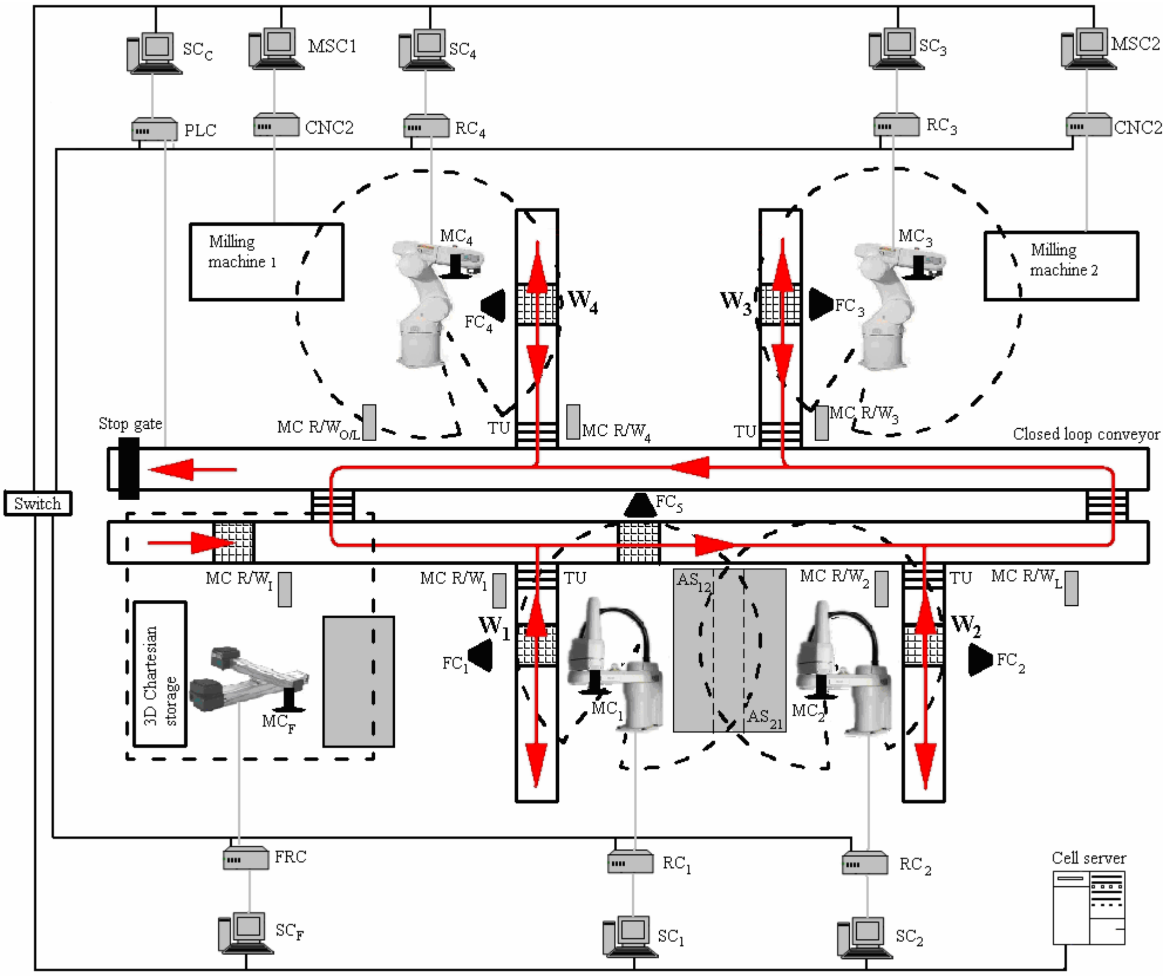

Figure 6. The production cell with networked robots and vision control.

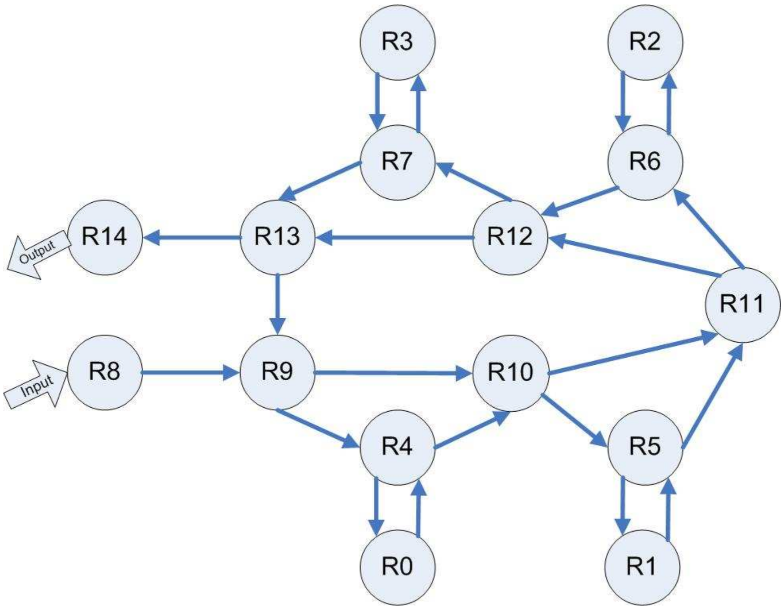

Figure 7. The manufacturing cell modelled as an oriented graph.

Fig. 8 shows the dependence of the total batch processing time as a function of the number of pallets (products) being in simultaneous execution at any moment of time (this number is constant, and must be a priori established by experiments). The time needed to compute a schedule for a fixed maximum number of pallets within the system is less than one second. The job scheduling program generating expertise holons (recommendations for the sequence of product orders) runs on an IBM PC Pentium 4 at 1,7GHz, making thus possible job scheduling in a real time scenario.

For a small production, i.e. with a number of order holons comparable to the maximum number of pallets coexisting into the system it can be seen that a minimum time is achieved for a maximum number of 6-7 palettes simultaneously present in the system. If the production grows, the total processing time decreases with the number of the pallets coexisting in the system (Fig. 8). This decrease depends on the product type and the distribution of operations on the resources, but if more than 6 palettes are simultaneously in execution, the risk that the conveyor gets blocked increases. This happens because there might be pallets on the main belt waiting to enter workstations (W), and pallets with products just processed and leaving the stations to re-enter the main belt (C) at the same moment of time.

The results presented in Fig. 8 confirm the correctness of the job scheduling algorithm producing expertise holons (off line recommendations of product sequence) relative to the class of transportation system in Fig. 6.

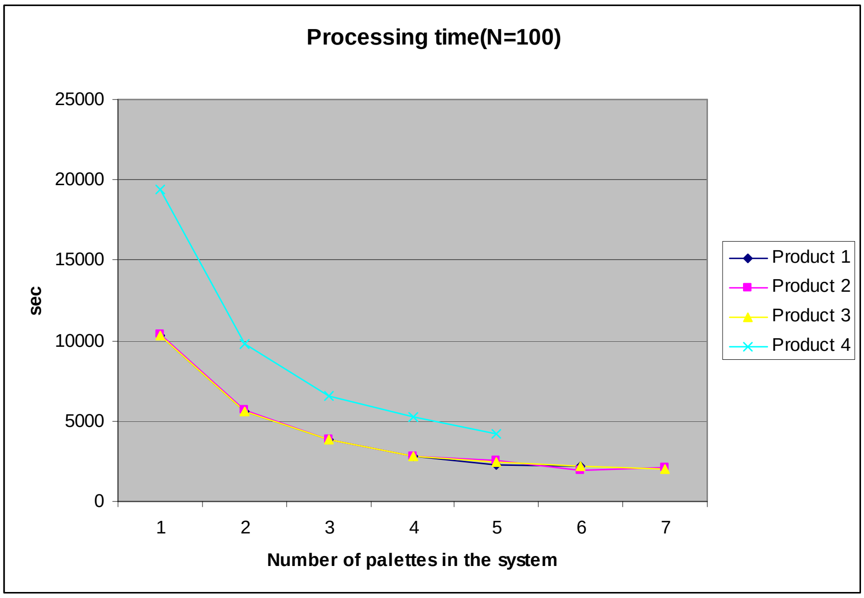

Figure 8. Variation of the batch processing time relative to the number of pallets in simultaneous execution.

REFERENCES

Anton, F.D., Borangiu, Th. and Silvia Anton (2007). An Implementation of High Availability in Networked Robotic Systems, Proceedings Vol. of the 4th Int. Conf. on Informatics in Control, Automation and Robotics – ICINCO 2007, May, 9-12, Angers.

Babiceanu, R.F., Chen, F.F. and R.H. Sturges (2004). Framework for control of automated material-handling systems using holonic manufacturing approach, Int. J. Prod. Res., Vol. **42**, No. *17*, 3551-3564, Taylor & Francis.

Barata, J (2005). Coalition Based Approach for Shop Floor agility, Amadora – Lisbon, Orion Edition.

Bongaerts, L., Wyns, J., Detand, J., Van Brussel, H. and P. Valckenaers (1996). Identification of Manufacturing Holons, Proceedings of the European Workshop for Agent-Oriented Systems in Manufacturing, Berlin, September 26-27, eds. S. Albayrak and S. Bussmann, 57-73.

Bongaerts, P., Monostori, L, McFarlane, D. and B. Kadar (1998). Hierarchy in Distributed Shop Floor Control, Proceedings of the 1st Int. Workshop on Intelligent Manufacturing Systems IMS-EUROPE, Ed. EPFL, Lausanne, 97-113.

Borangiu, Th. (2004). Intelligent image processing in Robotics & Manufacturing, Romanian Academy Publishing House, Bucharest.

Borangiu, Th., Balibrea, L.M.T., Anton, F.D., Ivanescu, N.A., Tunaru, Silvia, Dogar, Anamaria and S. Raileanu (2007). Holonic fault-tolerant control of networked robot-vision assembly stations, Preprints of the IFAC Workshop Intelligent Assembly and Disassembly - IAD 2007, May 23-25, Alicante.

Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L., and P. Peeters (1998). Reference Architecture for Holonic Manufacturing Systems: PROSA, Computers in Industry, Special Issue on Intelligent Manufacturing Systems, Vol. 37, (3), 255 – 276.

Cheng, F.-T., Chang, C.-F. and S.-L Wu (2006). Development of Holonic Manufacturing Execution Systems, Industrial Robotics: Theory, Modelling and Control, Advanced Robotics Systems, Vienna, 952 pp, ISBN 3-86611-285-8, published by Pro Literatur Verlag Robert Mayer-Scholz Germany.

Deen, S.M. (2003). Agent-Based Manufacturing – Advances in Holonic Approach, Springer.

Duffie, N.A. and V.V. Prabhu (1994). Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems, Journal of Manufacturing Systems, Vol. 13, No. 2, 94-107.

Iwamura, K., Taimizu, Y. and N. Sugimura (2005). A Study on Real-Time Scheduling for Holonic Manufacturing Systems – Simulation for Estimation of Future Status by Individual Holons Knowledge and Skill Chains, Engineering and Manufacturing Information Infrastructure in the Era of Global Communications, pp. 301-312.

Kusiak, A. (1990). Intelligent Manufacturing Systems, Prentice Hall, Englewood Cliffs, N.J.

Lastra, J. and I. Delamer (2006). Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap, IEEE trans. on Industrial Informatics, Vol. 2, No. 1, pp. 1-11.

Leitao, P. and F. Restivo (2006). "ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control, Computers in Industry, Vol. 57, No. 2, pp. 121-130.

Leitao, P., Mendes, J., Colombo, A.W. and F. Restivo (2007). Reconfigurable Production Control Systems: Beyond ADACOR, Preprints of the IFAC Workshop Intelligent Assembly and Disassembly - IAD 2007, May 23-25, Alicante, Spain, pp. 89-94.

Markus, A., Vancza, T. and L. Monostori (1996). A Market Approach to Holonic Manufacturing, Annals of the CIRP, Vol. 45(1), 433-436.

Morel, G., Panetto, H., Zaremba, M. and F. Mayer (2003). Manufacturing enterprise control and management system engineering: Rationales and open issues, IFAC Annual reviews in Control.

Onori, M., Barata, J. and R. Frey (2006). Evolvable Assembly Systems Basic Principles, IT for Balanced Manufacturing Systems, Vol. 220, IFIP, Ed. W. Shen, Boston: Springer, 317-328.

Pétin, J.-F., Gouyon, D. and G. Morel (2007). Supervisory synthesis for product-driven automation and its application to a flexible assembly cell, Control Engineering Practice 15 (2007), 595-614.

Ramos, C. (1996). A Holonic Approach for Task Scheduling in Manufacturing Systems, Proceedings of the IEEE Int. Conf. on Robotics and Automation, Minneapolis, USA, 2511-2516.

Tanaya, P.I., Detand, J. and J.P. Kruth (1995). Holonic Machine Controller: a study and implementation of holonic behaviour to current NC controller, Cooperation in Manufacturing: CIM at Work (Preprint), Kaatsheuvel, The Netherlands, August 28-30, 375-396.

Valckenaers, P., Van Brussel, H., Bongaerts, L. and J. Wyns (1994). Results of the Holonic Control System Benchmark at the KULeuven, Proceedings of the CIMAT Conference (Computer Integrated Manufacturing and Automation Technology), October 10-12, Troy, NY, Rensselaer Polytechnic Institute, 128-133.

Woolridge, M. (2002). An Introduction in Multi-Agent Systems, John Wiley & Sons.

Wyns, J., Van Ginderachter, T., Valckenaers, P. and H. Van Brussel (1997). Integration of Resource Allocation and Process Control in Holonic Manufacturing Systems, Proceedings of the 29th CIRP International Seminar on Manufacturing Systems, May 11-13, Osaka, pp. 57-62.

http://hms.ifw.uni-hannover.de, HMS Consortium Web Site.

www.rvholon.cimr.pub, RVHLON Web Site.