U.P.B. Sci. Bull., Series C, Vol. 73, Iss. 2, 2011 ISSN 1454-234x

DEVELOPMENTS IN THE HOLONIC CONTROL OF PRODUCTION FLOW

Claudiu PLĂIŞANU[1], Dănuț CLAPA[2]

Cerințele pietei şi presiunile din cadrul social necesită sisteme de fabricație eficiente care să țină pasul cu tendințele mereu crescânde din domeniu. De aici rezultă necesitatea dezvoltării unor sisteme noi de control al fabricației care să poată face față schimbărilor şi perturbațiilor din producție într-un mod eficient. În acest sens, în ultimul deceniu au fost făcute numeroase cercetări şi dezvoltări în domeniul Sistemelor de Fabricație Holonice. Lucrarea îşi propune să prezinte o metodologie de trasabilitate a fluxului de materiale bazată pe tehnici de control holonic, dezvoltând arhitectura de referință pentru sistemele holonice – PROSA. Este descrisă şi o aplicație (bazată pe tehnologii web), care are scopul de a-i ajuta pe utilizatori să conducă şi să monitorizeze la distanță procesul de fabricație.

Market requirements and social pressures require effective production systems that keep up the ever growing trends in the field. Hence, there is a need for developing innovative manufacturing control systems capable of coping with production changes and interferences in an effective manner. In this regard, the last decade has brought various research and developments in the field of Holonic Manufacturing Systems. This paper will present a methodology for the control and monitoring of the production flow, based on holonic control techniques and expanding the reference architecture for Holonic Manufacturing Systems - PROSA. Furthermore an application designed to help users remotely control and monitor the manufacturing process (web based) is being described.

Keywords: holonic manufacturing system, material handling, shop floor control, web monitoring

1. Introduction

During the last decade, three trends are challenging manufacturing systems: increasing product complexity, increasing product variety on reducing time-to-market and expansion of supply networks [1].

Enterprises no longer dominate the entire product life cycle and there is no enough time to develop the entire product alone. Hence, enterprise should be aware of concepts like concurrent engineering, lean production, core business,

[1] PhD student, University POLITEHNICA of Bucharest, Romania, e-mail: plaisanuc2001@yahoo.com
[2] PhD student, University POLITEHNICA of Bucharest, Romania e-mail: clapa1d@yahoo.com

virtual and federation enterprise. These new organization concepts lead to new approaches in production and therefore in Manufacturing Systems.

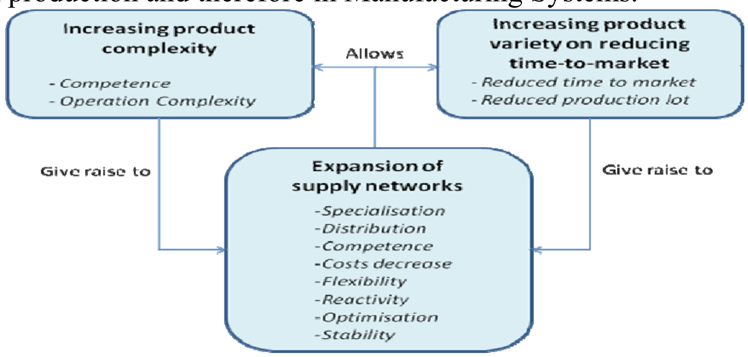

Fig. 1. Consequences and causes relation between trends [1]

Production distributed over several entities, complexity and high variation provoke challenges in manufacturing, namely in design, process planning, scheduling and control. Consequently, distribution, autonomy, flexibility, reactivity, cooperation and coordination mechanisms are issues that manufacturing system should deal with.

A manufacturing system capable of efficiently coping with such challenges requires a control system flexible enough to be controlled without significant delays and capable of increasing/decreasing products variety and also the production capacity. All these actions should take place without major interferences with manufacturing processes and apart from working environment variations.

All modern automatic manufacturing systems require a material handling system capable of efficiently transferring materials during the entire production process. In a modern production system, materials handling involves not only raw material and end products transportation and storage but also the identification, localization, protection measuring and materials control during processing, assembling and transport. This allows the integration of machine tools and other equipment that assure a continuous flow for the entire manufacturing process. Therefore, an efficient robotized system for materials conditioning is essential for most of manufacturing systems.

Since the control of materials handling machines represents an important component of entire manufacturing control system, the existing control architectures are used not only to control processing machines but also for controlling equipment that transfer materials between working stations (robots, AGV, etc).

Further on, in section 2 we'll make a comparison between the main manufacturing concepts and we present the holonic paradigm focusing on the most important holonic control architectures and in section 3 we are presenting an implementation of a holonic manufacturing platform with multiple robots emphasizing the control aspects of the production flow.

In section 4, a web application will be described for communication with global scheduler and for monitoring of the manufacturing cell.

2. Control Architectures for Holonic Manufacturing Systems

The recent literature on manufacturing control and materials handling defines four main types of control architectures [2]:

- Traditional control architectures – it is a centralised system there is only one control unity that makes all the decisions regarding materials flow into the system.

Advantages: simple architecture, has the possibility of global optimization. Disadvantages: long responses in high number of resources, lack of fault tolerance.

- Hierarchical control architectures – represent a top-down approach (master-slave), in which the commands flow is transferred strictly from high level controllers to low level controllers. Advantages: short responses, allow global optimization. Disadvantages: difficult to operate changes in the system, lack in fault tolerance.
- Hybrid control architectures – mainly derive from the hierarchical architecture, the difference being that this type of architecture allows the cooperation and sharing of information between low level controllers. Since the involvement of low level controllers in the decision making process is limited, the hybrid control architecture has the advantages but also the disadvantages of hierarchical control architecture.
- Hierarchical control architectures - is formed from a group of independent entities named agents that bid for tasks taking into account their state and future loading. In this case the master-slave relation does not exist, as in the above presented architectures. When the winning agent accomplishes the task it automatically becomes the new manager of the following task. Due to a decentralized architecture, the agents have total local autonomy and the system van properly react at any intervening change. However, since a job behavior depends on the number and characteristics of other commands within the system, it will be impossible to impose control optimization at a global level and the systems performance cannot be forecast.

The control of and scheduling in manufacturing systems corresponds to a distributed problem from the physical and from the logic point of view. Physically, the manufacturing system involves several resources (numeric control machines robots, AGV's, conveyors, or different entities with different resources) and from the logical point of view we are also dealing with a distributed problem. Due to these reasons for the control and dynamic scheduling of industrial tasks, hierarchical control architecture is best suited.

On the other hand, the aspect of the adaptation and flexibility in production systems will create the necessity for new and innovative control systems for manufacturing, such as: bionic manufacturing [5], genetic manufacturing [6], the fractal factory, random manufacturing [7], virtual manufacturing [8], and holonic manufacturing [9]. The table below compares all these manufacturing concepts:

Table 1

Comparison between manufacturing concepts [1]

	Bionic Manufacturing System	**Fractal Factory**	**Holonic Manufacturing System**
Entity	Cell	Fractal	Holon
Group	Organ	Fractal	Holarchy/Intermediate stable forms
Autonomy	Response to environment changes	Vitality through Adaptability	Limited by Canon
Coordination	Enzymes (coordinators)	Status exchange through	Conversation
Goals	Genesis & Environment data exchange	Globals, Inheritance and Dynamic formation	Top-Down Specialization
Plan & Control	Reactive to situations	Definition through iteration	Dynamic and Concurrent through communication
Main Focus	Reactive and Flexible	Multi-dimension approach	Background Concepts & Natural adaptation

The architecture that was approached and extended for the control of the manufacturing platform presented in the article is based on the reference architecture PROSA[4, 10].

This architecture mainly consists of three types of basic holons: order holons, product holons, and resource holons, each of them being responsible for one aspect of manufacturing control (technological planning, resource availability and logistics)[11].

Being a reference architecture, PROSA offers only general guiding marks for the system architecture. It refers to the material handling system as being a resource holon, but it doesn't specify what functions are fulfilled by each base holon when a transport operation is required[3,12]. In PROSA architecture, a resource holon contains a production resource of the manufacturing system (physical part) and an information processing part that controls the resource. This kind of holons offer production capacity and functionality to the surrounding holons.

The process and product information is maintained in a product holon assuring the quality of the product to be manufactured. Other information that a product holon contains can be: product life cycle, user requirements, design, process plans, bill of materials, quality assurance procedures, etc. An order holon represents a task in the manufacturing system. This kind of holons will perform the work that has been assigned to them in a correct way and in a timely manner.

3. Holonic control of the production flow

Production flows are modular at the moment, which means that each module within an enterprise is specialised and utilised for a certain task. In numerous cases the modules are connected and the materials are sequentially processed in each module, finally resulting a unique product or assembly.

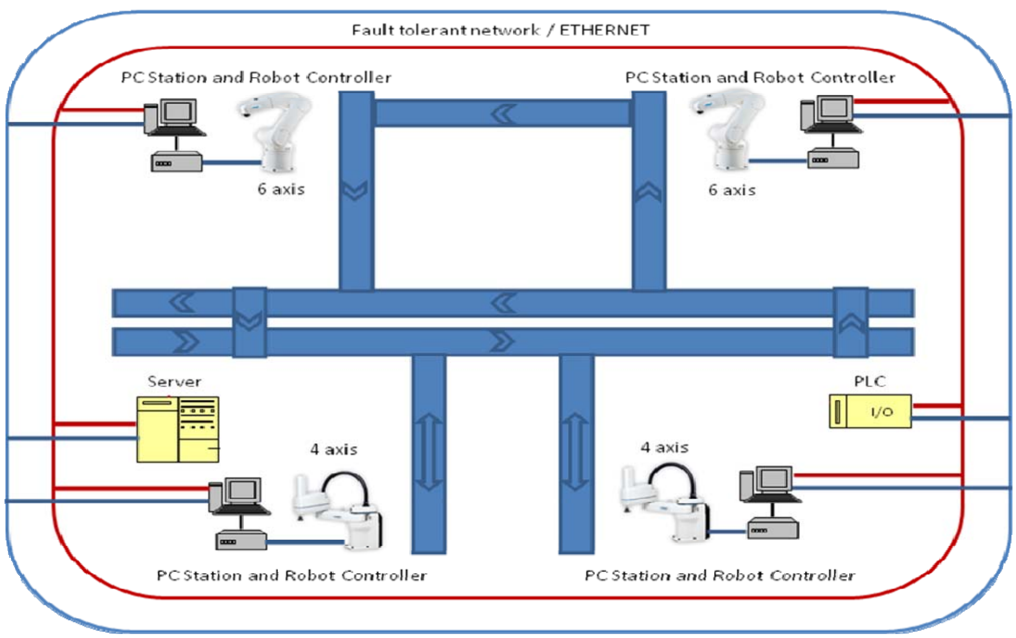

Fig. 2. The physical structure of the multi-robot platform

The physical structure of the multi-robot platform presented within this paper together with the related control mechanisms are developed according to the holonic manufacturing concept and it is presented in the figure 2. The platform contains four robots (vertical and horizontal articulated) with access to a closedloop and double routed conveyor for parts' transportation. The transfer of parts from the internal to the external band of the conveyor is controlled by a programmable logic controller (PLC) directly accessible for editing and downloading programs from an IBM PC computer. Robot controllers are master units of the four working stations, communicate together and are connected in a fault tolerant multi-network with the related computers.

The innovative approach proposed in this paper is the extension of the traditional control of materials to materials conditioning functions, permitting in this way to relax the constraints imposed on materials flux and transport means but also the real-time integration of quality control in the manufacturing global system. This functional extension becomes possible by applying Artificial Intelligence concepts in order to obtain a global autonomous behavior, self-training, depending on context and tasks of the manufacturing process and adaptable at the working environment.

In order to achieve this goal, an innovative holonic control architecture was approach for the presented manufacturing system. Based on PROSA reference architecture, we proposed the the following entities for the manufacturing control system: Command (Control) Holons (production order), Resource Holons (for material handling), the global scheduler, the monitoring entity and system's database, materialized through a control and computing structure on three levels, interlinked by a fault-tolerant communication network that follows the execution of the manufacturing tasks [13].

Having more than one control unit, the proposed architecture differs from traditional centralized control architectures that have only one control unit for all of the manufacturing process. The information flow between control units is always bidirectional because of the decentralized structure of the architecture as well as the applying of holonic concepts of cooperation.

In the holonic platform, a control holon represents a task or job (a production order) and all of its associated information, incorporated in one control unit, while the resource holons are represented by the physical processing resources, each having its own control unit.

There can be three types of resource holons – machine, handler and equipment.

The machine holon is made up of all of the processing devices in the system, together with their control units, and it is responsible for the physical data regarding the parts – from raw materials to finished products.

The material handling holon (handler holon) is made up of all transport systems, robot systems, together with their control units and has as responsibility the transport of raw materials, of the half-finished materials and of the finished products within the manufacturing cell.

The equipment holon contains all the equipments in the manufacturing system, other than processing machines and transport equipments represented by material handling holons. The examples considered in the research activity are: measuring devices, intermediary buffers, quality control systems (such as artificial sight systems) and conveyor belts that transport the products in the working area of the robots or outside the manufacturing cell after the processing is finished.

The global scheduler computes the planning sequences; however, these are treated as advices by the decision entities.

Considering that the objective is to control the hardware devices and equipments (robots, conveyors, etc) which are handling and conditioning the materials (row materials and components) within the manufacturing system, the holarchy can be simplified according to the architecture presented further on.

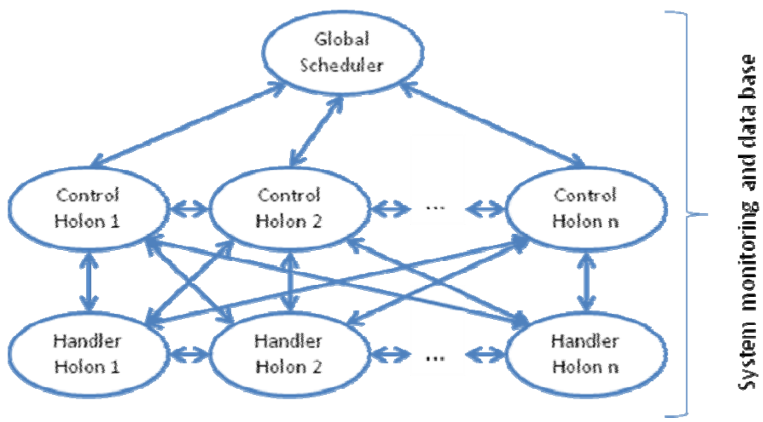

Fig. 3. Holonic Architecture for Material Handling and Conditioning

The global scheduler computes the planning sequences for the material handling and conditioning devices, as it keeps a global image over the entire manufacturing system; however, these are considered only as “advices” by the other decision entities within the holarchy.

The control holons are entities that have the authority to designate transport tasks to the handling holons, every time a new task (production order) enters the system.

The material handling holons (handler holons) are made up of the physical robot handlers together with their controllers, conveyor and PLC. Their main tasks are the execution of all of the conditioning activities of the materials (recognition, localization, qualifying, transport, handling, feeding the machinery) in the manufacturing system. Having its own control units, the handling holons can decide the course of its own actions on the basis of their status and their future loading. Based on autonomy and cooperation – characteristics specific to a holonic system – the handling holons can make changes in the existing plans when one of them is facing an important, unforeseen, loading or when it requires maintenance. Predefined cooperation methodologies and protocols will be used between the handling holons (robot holons) in order to achieve these modifications, without any intervention from the control holons [13].

The system's monitoring and data base module has no decision authority in the way the tasks are executed; it operates as data base for already executed jobs and monitors the availability of material handling devices in the system and transmit the new jobs or signals to stop or start production. This module communicates with all the entities in the system and transmits to the control holons and global scheduler information about that availability. The control holons must determine which handling holons are available in the system in order to establish to whom the new transport requests must be sent for the preparation of the offer, while the global scheduler must establish which handling holons are available for the calculation of the adequate planning sequences and adequate task assignment. This module has also other functions, such as: creation of control holons every time a new production order (command) enters the system, the canceling of a control holon when all of the transport tasks associated with it are over, updating the data base with information regarding completed tasks for future references. The material conditioning operations are executed by the handling holons after an algorithmic evaluation of the transport requests from the control holons. The individual planning sequences from the handling holons and global scheduler are evaluated by the control holons and the transport tasks are assigned to the agents, choosing the received schedule with the lowest cost. The final decision is taken by the control holon after it receives in the allocated time the transport offers from the entities in the system. A control holon must hold sufficient information about the next job and must send it to the handling holons and global scheduler so that the next transport process is made possible. For each physical resource (transfer systems of the conveyor, actuators, sensors, etc) available in the system, representing a holon, there were designed sequential control algorithms which allow material transport and conditioning in the holonic manufacturing cell [14]. In this respect, for each new task, there are dedicated holons with specific responsibilities (including the right for taking the final decision regarding the material handling, conditioning and transportation within the manufacturing cell) which will be able to control the related resource.

The final decision is taken by the related control holon based on the "offers" received from the available handler holons and the global scheduler. The process of task allocation is presented in the figure below:

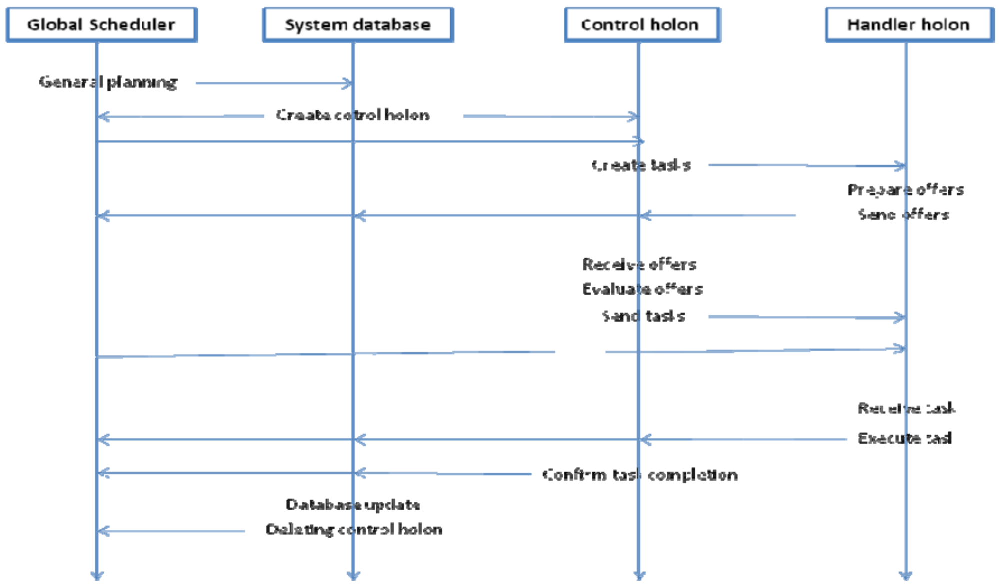

Fig. 4. Task allocation within the developed holonic manufacturing system

Each resource has its own control routine that manages the received production orders, verify the availability of system's holons and makes decisions to transport the product further on in the system or to transfer it to the holons representing the robots. With the help of OPC, data is received from the global scheduler.

4. Web monitoring application and communication with global scheduler

In order to help users to remotely control and monitor the manufacturing process, a web application is being considered. The application for web monitoring of the holonic platform is completely modularized, and it is weakly linked with the global scheduler. According to this logic, the two applications behave like two entities using a communication protocol in order to exchange messages. To better illustrate this, the picture below described the application diagram, and later on the communication diagram between these applications will be detailed.

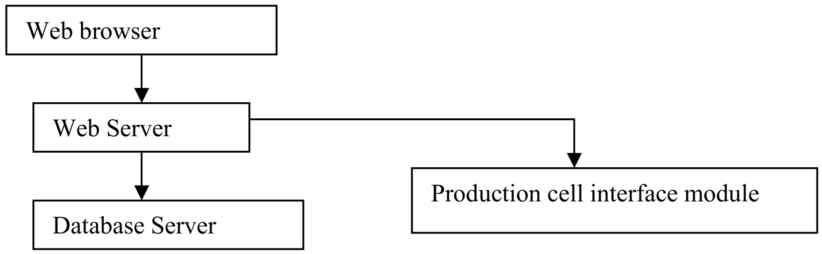

Fig. 5 – Web monitoring application architecture

In the architecture presented in the picture above, the web browser could be any regular browser, such as Internet Explorer, Firefox, Opera, etc; the web server is the server where the user interface will be installed, the database server is the server which will hold all the data regarding orders, products etc and the production cell interface module represents the communication interface between the global scheduler and the user interface.

Here we can see that the method through which we submit operations for execution to the production cell is fully transparent for the user. The user is not forced to have advanced computer skills. We can also see that this structure is perfectly scalable because we can change the database server at any moment without interfering with the activity of the production cell. Any modification made by the application architects or by the application itself are again fully transparent for the user, on the one hand, and for the production cell on the other.

To illustrate this modularization, the picture below shows the components of the application.

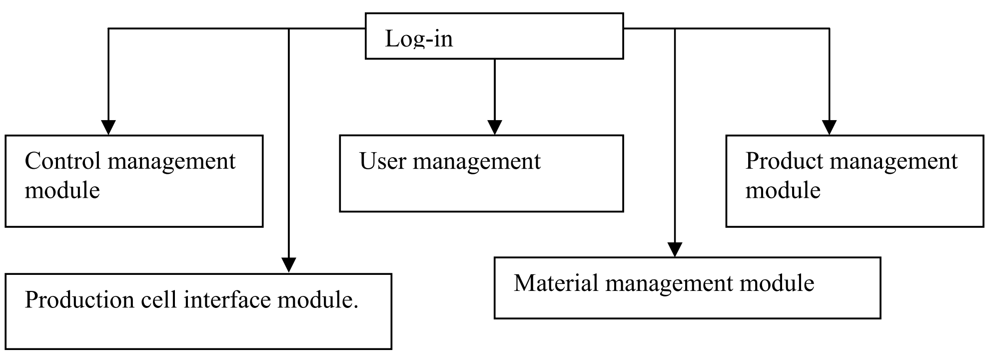

Fig. 6. Web monitoring application components

- *Log-in module* is used to manage user activity and access zones depending on their permission levels.
- *Command management* module represents the module responsible for the monitoring the execution of commands issued by the clients and

from there is the production controlled by starting, stopping or delay execution of the products (this command are transmitted trough the command file described below in the “a. planner input data” section) .

- *User management* – is the module where the data of registered users can be modified.
- *Product management module* – the module where products are virtually created from raw materials based on the available stock.
- *Material management module* – the management module for raw materials required for the development of products.
- *Production cell interface module* – is the communication module with the software installed on the production cell.

In the diagram above we can see that the application is made up of modules communicating through messages transmitted from one module to the other or archived in the database for usage for other modules. For instance, we can change the login module without interfering with the user management module, which does not usually happen since there is usually a tight connection between these two modules. Most of the times application architectures are built to incorporate these two modules in either a single one or in two strongly linked modules.

Further on, we are going to present the communication protocol between the client interface and the production cell management application. The production system is made up of the elements listed below, who will exchange the following information (messages),:

The client interface and the cell planning and management application will be synchronized through a data exchange through buffer files. Communication is carried out through messages sent by these files.

a. Planner input data

1. “input_nr_order” file – for a command the client will provide the producer with the following information: product type, quantity and priority. There are 4 priority levels (0, 1, 2 and 3) where 3 represents the highest priority commands. Once this information is submitted, a connection must be made between them and the actual production, in order to be able to report the command status to the client. Therefore, besides the 3 fields defining a product command, another field is added making the connection with the command client, index client.

Thus, the file will have the following line structure:

nr_products_order$priority$product_name$client_index

Note: “nr_order” (within the “input_nr_order” file) is an integer that grows for each command and resets when it reaches the limit. For each planning-production cycle, the planning application will take the lowest index file and delete it.

2. "command" file – it is recommended that, once the orders are put into production, there is a way to delay or cancel their execution, for two reasons: new commands have been received with an order of priority exceeding the queued commands, who are not urgent, or which, for whatever reason (such as command is no longer required) can be canceled. This is why, this single-line file (it contains one single command) is used to send messages to the planner, who interprets them as follows:

- *start production* – means that the planner can start production. It is a standard for the command file to contain this command;
- *stop production* – means that the planner must stop production and cancel all commands, or if, any command is listed to be canceled specifically, it cancels only the specified commands;
- *express command* – means stop production, read a new entry (input_nr_order file) and then plan and execute the entire batch.
- *cancel product* – means reconfigure the planning if execution of the remaining products and the product will not be planned to be executed till the administrator send it to execution (this is possible obviously only if the product isn't already executed or in execution).

b. Output planner data

1. "feedback" file – This is a file containing the command status. It has the following line structure: product_type, execution_start_time, execution finish_time, index_client, status, with the following meanings:

- *product_type* – the type of the product specified in this file
- *execution_start_time* – the time when the product production started/will start
- *execution finish_time* – the time when the product production will end/ended
- *index_client* – the index monitoring the products of certain clients (for instance, if there are two clients who ordered the same product, they must know which is which)
- *status* – it expresses a product status in the following manner: failed_execution (the product can no longer the produced due to a lack of raw materials or resources), failed (the product can no longer be produced due to a malfunction occurring during the manufacturing process), processing (undergoing processing) and done (the product is complete). If the status is failed, then it will also mention the malfunction or error to allow for future evaluation, in order to anticipate any possible malfunction through specific application algorithms (for instance data mining algorithms, linear algorithms etc.)

Note: On the basis of the two times above and of the global system time, the application knows is the command was carried out, is being carried out or is about to be carried out. This file will be updated once a minute.

When a batch is complete, there will be two types of products: complete and failed. Note that for each event recorded by the system (breakdown/lack of resource) planning is redone, taking into consideration all products previously marked as compromised due to the lack of resources. Failed products have been given this status for 2 reasons: a malfunction was recorded on the production line or no raw materials were present. For this reason the planning and management application sees to it to re-process one last time products failed due to production flaws. If the product manufacture remains incomplete, then a system flaw is considered to be the cause for this (for instance: a part is misassembled because the pallet is not perfectly in line with the robot, this is identified by sight, but the product remains catalogues as discarded).

2. “Lock” file – this is a temporary file that expresses the planner usage as follows: if it exists, it means that the cell planning and management application is busy with a previous command, and if it doesn’t exist, a new command can be sent to the planner. This file contains the date and time when the system was locked.

c. Production information archiving file (log)

1. “output_time_data” file – After a batch is produced, it is recommended that the resulting information is stored for future processing. Thus all information regarding product traceability will be stored: operations, production times, production input times, operation resources and finally product output status. This can be catalogued as done or failed. All this information will be archived in communication files until they are processed by the application and entered into a database for evaluation; the name of these files will begin with “output” and will be followed by the time and date when the file was first written.s

2. “error.txt” file – all system malfunctions will be stored in this file as records, such as:

- *Data*: the date when the error occurred;
- *Time*: the time when the error occurred;
- *Error type*: a string of characters describing an error (e.g.: P-type product – failed, and the operations that resulted in this failed product);
- *Note*: the error types will be deterministic, meaning that the same error codes will be written for each error (type).

All these files can be found on a shared disk as follows:

- The root will contain the lock inter-blocking file, the command file, error.txt for storing errors, and the feedback file.

- The Input directory, located directly in the root, will contain the product request files.
- The production already carried out will be recorded in separate files for each batch, and the files are grouped into directories by days.

The architecture described above can experience issues with the communication procedure, such as: one of the files can get stuck in the write-to status by one of the application, thus blocking the access of the other. This issue is solved as follows: the second application will wait for the resource to become available for awhile, and if it doesn't it will write the information in a file with the same name plus the prefix "bak". If the issue persists or is repeated more than three times the administrator is warned in order to troubleshoot and solve it.

The file blockage issue is solved by a communication protocol such as TCP/IP, but in this case the communication issue occurs between the applications, when one of the communication interfaces remains busy for a time exceeding the "time out" period set for the application, or even ceases to respond for various reasons. In this case, either the application submitting a request attempts the re-establish communication until it succeeds, which results in application blockage, or it cancels the request and the information is stored to be submitted when communications are back or the information is lost. The advantage of this type of communication is that the shared disk is not excessively used. The disadvantage is that information can be lost due to the communication network between the applications.

5. Conclusions and future developments

The manufacturing platform presented in this paper was being developed under a national research project as a pilot application which was implemented and tested in the Laboratory of "Robotics and AI" of the University Politehnica of Bucharest. In essence, the area of applicative and fundamental research proposed in the project will contribute to the development of a high performance infrastructure in manufacturing field and to the concentration of a significant mass of human resources with educational purposes.

The main goals of this project was to highlight the potential of the control system to be capable, both in part and as a whole to conduct manufacturing in an environment where change is frequent and unpredictable

Regarding the future developments, we can say that there are at least two critical issues that must be addressed before holonic control solutions can be expected to play any significant part in next generation manufacturing systems:

- The combination between the production cell and the web monitoring application for this production cell creates the possibility of a factory fully automated

- *Establishing Suitable Implementation Approaches with Existing and Future Commercial Computing Systems:* From an implementation perspective, there has been little work done in determining the compatibility of the holonic vision with the current or the next generation of industrial control and computing systems. Holonic systems will require a high level of reasoning and computational capability at the shop floor levels, coupled with more flexible communications and more dynamic interfaces to human operators and users. Determining how to construct and interface systems capable of fully supporting holonic operations with existing legacy systems will also be a major issue as holonic systems capabilities reach industrial strength. In the shorter term, suitable migration approaches for the implementation of intermediate holonic control capabilities are required, involving the combination of PC, Programmable Logic Controller (PLC) and Machine (robot) controller to provide a holonic control infrastructure for an assembly cell, like the one presented within this paper.
- The system's functionality is fully modularized, and if for any reason the web application is down the production cell can continue to operate, as it is not directly dependant on this interface. Moreover, if the production cell is down for any reason the web interface will continue to operate

REFERENCES

[1] *N. Silva, P. Sousa, C. Ramos*, A Holonic Manufacturing System Implementation, Proceedings of the Advanced Summer Institute (ASI'98). Bremen, Germany, 1998

[2] *D.M. Dilts, N.P. Boyd, H.H. Whorms,* The evolution of control architectures for automated manufacturing systems, Journal of Manufacturing Systems, 10, pp. 79-93, 1991

[3] *A.H. Overmars, D.J. Toncich,* Hybrid FMS control architectures based on holonic principles, International Journal of Flexible Manufacturing Systems, 8, pp. 263-278, 1996

[4] *P. Sousa, C. Ramos,* Scheduling Orders in Manufacturing Systems using a Holonic approach, European Workshop on Agent-Oriented Systems in Manufacturing, 2001

[5] *N. Okino, Ed.J Peklenik ,* "Bionic Manufacturing Systems", in Flexible Manufacturing Systems, Past, Present, Future, CIRP, Faculty of Mechanical Engineering, Ljubljana 1993, pp. 73-95

[6] *K. Ueda, Ed.J Peklenik,* "A Genetic Approach toward Future Manufacturing Systems", in Flexible Manufacturing Systems, Past, Present, Future, CIRP, Faculty of Mechanical Engineering, Ljubljana 1993, pp. 221-228

[7] *K. Iwata, M. Onosato,* "Random Manufacturing System: a New Concept of manufacturing systems for production to order", Annals of the CIRP, **Vol. 43**/1/1994, 1994, pp. 379-384.

[8] *F. Kimura,* "A Product and Process Model for Virtual Manufacturing Systems", Annals of the CIRP, **Vol. 42**/1/1993, 1993, pp. 147-150

[9] *H. Van Brussel,* "Holonic Manufacturing Systems, the vision matching the problem", Proc. Of First European Conf. on Holonic Manufacturing Systems, Ed. IFW-Hannover, Hannover, December 1, 1994
[10] *Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, Patrick Peeters,* "Reference Architecture for Holonic Manufacturing Systems: PROSA", Computers In Industry, **Vol. 37**, 1998, 255-274.
[11] *A. Koestler,* "The Ghost in the Machine", Arkana Books, London, 1969
[12] http://hms.ifw.uni-hannover.de/
[13] *Th. Borangiu, F.D. Anton, S. Tunaru, A. Dogar,* A Holonic Fault Tolerant Manufacturing Platform With Multiple Robots, 15 th International Workshop Robotics in Alpe-Adria-Danube Region RAAD 2006, June 15-17, Balantonfured, Lake Balanton, Hungary, 2006
[14] *C. Plaisanu, D. Niculae, R. Sora, L. Cerban, D. Clapa, A. Enache*, Control Techniques for a Holonic Manufacturing Platform with Multiple Robots, 16th edition of the 2008 IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2008.