Intelligent Manufacturing Systems: a methodology for technological migration

Jorge Gamboa-Revilla and Miguel Ramírez-Cadena. *

Abstract—**More complex products in efficiency and quality, the necessity of diminish energy spending and investment reduction; amongst other things, are disturbances, and their occurrence may have severe impact in the performance of actual manufacturing systems. Manufacturing systems should be based in distributed and autonomous entities, being possible the addition of new components without stopping or restarting processes. All these facilities point to the concept of agile manufacturing systems. The approach is addressed to encourage the usage of holonic and multi-agent concepts in traditional production lines, with a friendly software upgrade and a minimum cost in hardware expansion. A methodology that includes the technological migration from a established flexible manufacturing structure (FMS) to intelligent and reconfigurable manufacturing system (RMS) is presented. An example of implementation will be described in depth to show the viability of the proposed schema.**

Keywords: Computer integrated manufacturing, Distributed control, Holon, Intelligent manufactuting systems, Multiagent Systems, Parallel architectures, Parallel processing, Reconfigurable Manufacturing Systems

1 Introduction

In the last twenty years manufacture concepts have had several redefinitions, in the eighties, the concept of flexible manufacturing systems (FMS) was introduced to develop a new family of products with similar dimensions and constraints. But nowadays, the capacity of reconfiguration has become a major issue for improving the functioning of industrial processes. Indeed, today a main objective is to adapt quickly in order to start a new production or to react in a failure occurrence [1]. Intelligent manufacturing systems (IMS)[2], has both flexibility and reconfigurability, in fact this concept brings more than a few ideas of software intelligence meanings, which contemplates characteristics such as autonomy, decentralization, flexibility, reliability, efficiency, learning, and self-regeneration, all of these facilities lead to the concept of agent-based manufacturing systems. An agent is a computer system that is situated in some environment, and that is capable to act in an autonomous way in this environment in order to meet its design objectives. Intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their goals, this characteristic is well known as reactivity. However an agent is also proactive, for it agent is able to exhibit goal directed behavior by taking the initiative. In addition, agents are social, having the ability to interact with other agents [3]. It worth to remember the definition of an "Holon", which its similarities with agent definition, brings up controversial meanings, nevertheless an holon is well recognized on manufacture applications with the distinctive of a more specific intelligence use, while an agent could have different levels of intelligence such as logical, reactive, layered or in a more advanced way, with beliefs, desires and intentions (BDI)[4]. The word "holon" comes from the Greek holos that means whole, with the suffix on which, as in proton or neutron, suggests a particle or part. A system of holons that co-operate to achieve a goal or objective limited by rules of interaction is called holarchy [5]. On the past decade researchers have focused their investigations in the theory and design of holonic manufacturing systems (HMS), wherein can be found two principal aspects that at present are still being depurated. On one hand we have issues associated with the development of multi-agent systems (MAS), on the other hand how the MAS can be effectively deployed into manufacturing environments [6]. In spite of having a complete set of agent architectures and algorithms, they still do not have the strength to displace established manufacturing systems, even though the companies know that in a brief time market will change and some actions have to be taken. This paper presents a novel approach to manufacturing floor control design with agent coordination, including the interaction through a manufacturing execution system (MES) with manufacturing planning level (See fig.1) structure taken from previous researches [7, 8, 9]. This scheme uses commercial software that includes a few mainly distinctive characteristics, such as block oriented programming, parallelism for distributed structures, and the flexibility to scale platform capacities without missing the structure concept. This article will refer to a multi-agent manufacturing platform imple-

*Jorge Gamboa-Revilla, Miguel Ramírez-Cadena, Tecnológico de Monterrey (ITESM)Campus Monterrey, Mechatronics and Automation Department, Av. E. Garza Sada 2501, 64849, Monterrey, Nuevo León, México. (a00778197, miguel.ramirez)@itesm.mx

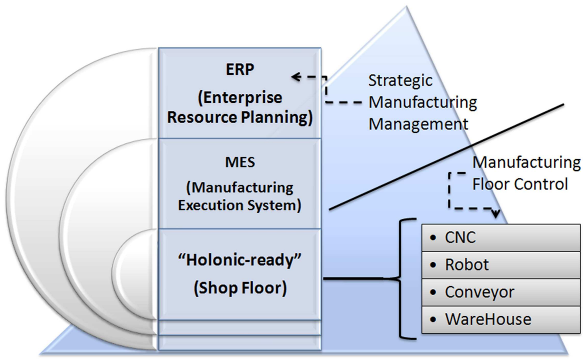

ERP
(Enterprise Resource Planning)
MES
(Manufacturing Execution System)
"Holonic-ready"
(Shop Floor)
Strategic Manufacturing Management
Manufacturing Floor Control
• CNC
• Robot
• Conveyor
• WareHouse

Figure 1: Floor control design with holonic coordination, including the interaction through a MES with manufacturing planning level.

mented at Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) as a general study case; nevertheless a methodology to convert conventional manufacture systems into new intelligent manufacturing, flexible and reconfigurable concept shall be explained in detail.

2 The feature of migration

Since the multi-agent technology has been recognized as a key concept in building a new generation of highly distributed, intelligent, self-organizing and robust manufacturing control solution, the traditional concept of manufacturing systems, has become vulnerable to changes [10]. Environmental changes, failure detection, reconfigurability, and expandability; are a set of capabilities that make an attractive option the application of this feature of migration. The idea of a standard software platform including characteristics such as reconfigurability, flexibility and "holonic-ready" [9] concepts, is justified by the necessity of uptake on established systems, making easier to adopt new production infrastructures without dramatic hardware changes and long setup times. At the moment it is possible to find several topologies of manufacturing cells, such as centralized, hierarchical, and heterarchical structures [11]. Each topology could be considered as optimal and able to accept migration, taking into account that each block should be related without complete dependency, at least after migration is implemented, and well functioning shall not be compromised with any other element from the cell. A generic platform was designed in order to apply multi-agent schema [9]. The platform design was implemented in such a way that any flexible manufacturing cell could be evolved into agent-based structure. The clue is to adopt the platform structure, and shape each element (robots, numeric control machinery, conveyors) of a cell to acquire agent personality. Once the problem or problems are identified the MAS design phase, starts, which is more oriented toward the implementation of the generic platform; however a methodology should be committed. The methodology includes definition of: 1st stage; capabilities of each single agent, and the inter-agent communication, 2nd stage MAS architecture planning.

3 The Methodology

Before any attempt can be made to implement agent societies effectively in a manufacturing system, an analysis of the industrial life cycle is pivotal. It therefore becomes important to introduce the environment in which an agent should act [12]. For it the information system of a manufacturing enterprise is crucial to be recognized, in order to clearly sketch how agents can be integrated and how data would be interchanged (See fig. 2), wherein the three layers, that computer systems in manufacturing management use, are illustrated. The generic platform is toward from general to particular application, so before start working on developing intelligence, is crucial to make independent each element, which is supposed to emerge from a centralized and sequential architecture that actually shall be substituted by the new platform.

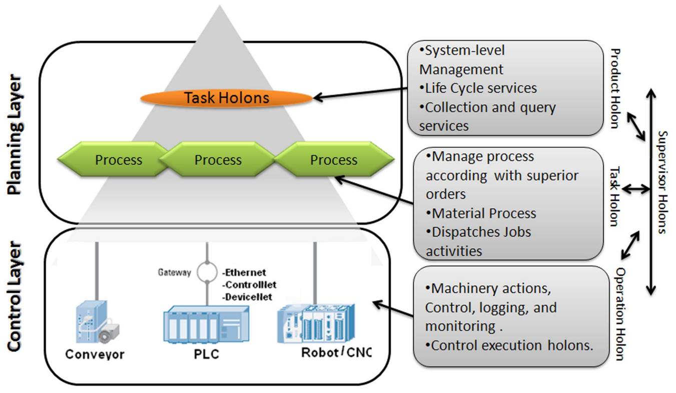


Figure 2: Hierarchy model of communication and interaction

This section will start taking the hierarchy presented on figure 1 and model presented on figure 2, taken from earlier works on this research [7, 8, 9]. The superior part of this pyramid is performed by management layer, which are satisfied with a manufacturing planning level, and a manufacturing execution level. Both could be programmable holons, purely software based. In addition pyramid bottom is formed by executable holons, which has direct contact with machinery, and hardware systems, also this part of the pyramid frequently is the one with more constraints in manufacturing environments. The efforts on this section will be driven to get the pyramid base prepared to be adapted without neither hardware changes nor design, on the other hand ready to become reconfigurable, and holonic-ready [9], the methodology (See fig. 3)shall be explained as follows.

- *Define Communication Structure:* This answers the issue, how data acquisition will be performed and how data shall be shared between items on internal and distributed network. On actual systems, this is an easy matter due the well-known kinds of communication protocols. Data interchange is available in several ways, for example ActiveX, data library functions (*.dll), OPC and data sockets, and those at the same time use different well established channels of communication, such Control-Net, Profibus, Ethernet, Device-Net, amongst others, making even easier this step on the methodology.

- *Isolate from global system:* The well or bad functioning of one element should not affect to the other elements functioning. In other words is essential to rupture dependences. Isolating, could be a difficult part on this methodology, understanding that each element in traditional systems is related with a strong sequential logic, using strategies such as First-in-first-out (FIFO), Earliest Due Date (EDD), Shortest Processing Time (SPT), and Least Slack First (LSF), all of them has an acceptable performance and their use have solved many industrial optimization problems, nevertheless are sequential dependant, sequential operations, and dependency are rigid and that structure is not compatible with the ideal holonic infrastructure.

- *Convert from general to particular:* Scalability and generic features are the main topics on this methodology; we must conserve generality, thinking in advance to future hardware or software changes. Rigid and dedicated operation should be eliminated, to achieve different applications, making able to change its functions.

- *Create relationships but not dependences:* Elements should be able at the end to share data, hence is necessary to establish a weak relation with messaging protocols such as FIPA or contract Net, that with fire actions in order to perform an application. It follows that relationships must be created without missing complete agent-based structure.

The result after this methodology would be what we call a "holonic-ready agent" (HRA), which meaning contemplates an entity with characteristics and attributes necessaries to adopt intelligence blocks (to become a Holon) in order to achieve different functions or tasks. An overall view of the resulting platform (See fig. 4) emerges from figure 4, where is shown in a more oriented way the methodology applied on the commercial software used to develop the generic platform. The methodology makes

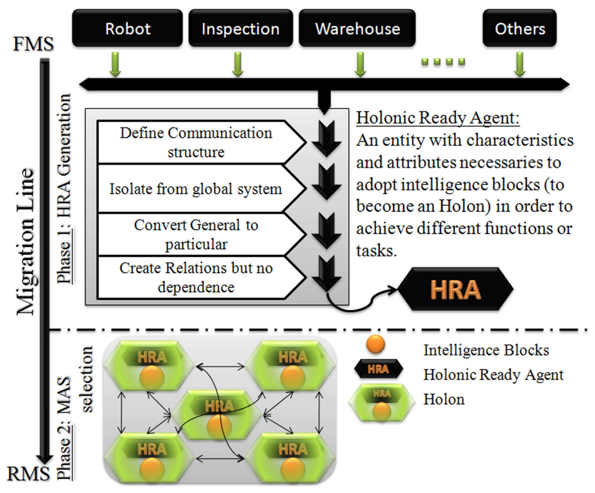

Robot
Inspection
Warehouse
Others
FMS
Migration Line
Phase 1: HRA Generation
Define Communication structure
Isolate from global system
Convert General to particular
Create Relations but no dependence
Holonic Ready Agent:
An entity with characteristics and attributes necessaries to adopt intelligence blocks (to become an Holon) in order to achieve different functions or tasks.
HRA
Phase 2: MAS selection
RMS
Intelligence Blocks
Holonic Ready Agent
Holon

Figure 3: The methodology model implemented to achieve migration feature

possible reconfigurability into the manufacturing cell, and at the same time the cell becomes ready to adopt second stage of the problem, Multiagent architecture selection. An to explain how reconfigurability is done, the robot routine, which contains ethernet procedures and *.dll functions to perform actions such as movements or execution of a specified routine. Lets imagine that we have to plug another identical robot to the cell, following this methodology procedure, it is just matter of duplicating robot cycle and change some kind of IP address to achieve plug and produce, like some authors have defined [13]. Holonic or intelligent agent skills and knowledge should be attached to the inputs and outputs of those isolated cycles, the intelligent entities could be developed on different coding resources, such as Matlab, C++ or JAVA, and these can be encapsulated as an external code node in such a way that they can be used on LabView interface.

4 Agent Architecture for generic platform

Before continuing with the study case, is fundamental to define some aspects about the agent architecture implemented, remember this refer to the second phase of migration problem. An agent could be categorized in purely reactive, in which do not consider historical data to react, and agents with state, this category contemplates past events, and contain internal states that describe the agent current situation, its perception of the world map to a set of possible actions to react in different manners. However these aspects still do not clarify how functioning might be implemented, functioning classes could be logical, reactive, intentional flexible (BDI) and layered, for this application case reactive agents is selected, in which decision making is implemented in some form of direct

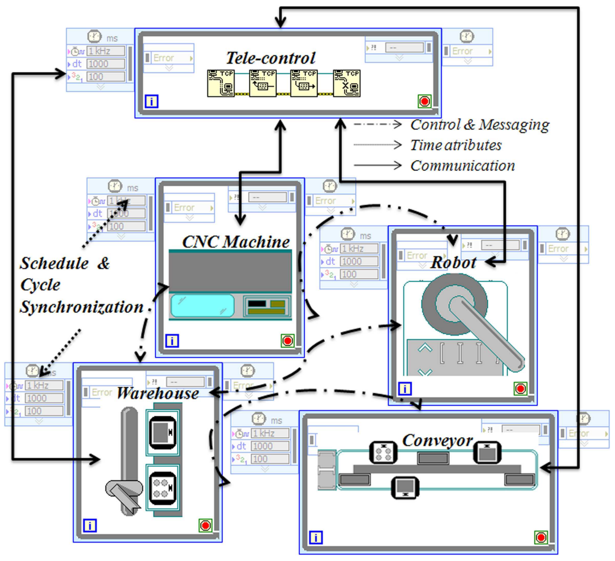


Figure 4: The distributed platform cycle design and message structure.

mapping from situation to action. For this implementation is necessary to create three different sets or vectors, in order to define the agent structure and its virtual environment. Equation 1 denote a representation of a set of discrete states **E**, which we can justify by pointing out that any continuous environment can be modeled by a discrete environment to any desired degree of accuracy, on the other side we have **Ac** being a set of discrete actions. The basic model of agents interacting with their environments is represented on equation 2, as can be seen **r** is a sequence with actions firing states, hence equations 3 and 4 are sequence terminated by actions or states respectively. The environment starts in some state, and the agent begins by choosing an action to perform on that state.

$$\mathbf{E} = s0, s1, ..., , su+1; \mathbf{Ac} = \alpha 0, \alpha 1, ..., , \alpha u + 1; \quad (1)$$

$$\mathbf{r} : s0 \Rightarrow^{\alpha 0} s1 \Rightarrow^{\alpha 1} s2 \Rightarrow^{\alpha 2} ... \Rightarrow^{\alpha u} su+1 \quad (2)$$

$$\mathbf{R^{Ac}} : s0 \Rightarrow^{a0} s1 \Rightarrow^{a1} ... \Rightarrow^{au} \quad (3)$$

$$\mathbf{R^{E}} : s0 \Rightarrow^{a0} s1 \Rightarrow^{a1} ... \Rightarrow^{au} su+1 \quad (4)$$

As a result of this action, the environment can respond with a number of possible states. However, only one state will actually result, and obviously the agent does not know in advance which it will be. The rules that govern environment are established by the state transformer, equation 5, at the same time each agent is defined by equation 6, in which an agent receives a run or sequence terminated by a state, an agent should map this situation to an action[3].

$$\tau(R^{Ac}) : R^{Ac} \Rightarrow \wp(E) \quad (5)$$

$$Ag : R^{E} \Rightarrow Ac \quad (6)$$

Although architecture is designed by these abstract models, the following pseudo code, represent in a very general way how these models are implemented and the study case is developed:

```
1. function action(p:P):A
2.   var fired:(R)
3.   var selected:A
4. begin
5. fired:={(c,a)|(c,a)~R and p~c}
6. for  each (c,a)~fired do
7. if !((c',a')~fired such that (c',a')-<(c,a))
8. then return a
9. end-if
10. end-for
11. return null
12. end function action
```

Thus action selection begins by fires computing the set fired of all behaviors that fire (line 5). Then, each behavior (c, a) that fires is checked, to determine whether there is some other higher priority behavior that fires. If not, then the action part of the behavior, a, is returned as the selected action (line 8)[14].

5 The ITESM manufacturing cell

The laboratory installed at ITESM, consist of two identical cells equipped with one loop belt-conveyor, one robot (Motoman UP6), one ASRS (automatic storage retrieval system) installed in a warehouse of 2x12 storage slots, a CNC machine(EMCO PC MILL 155), and an assembly table for each cell (See fig. 5). The conveyors have three docking stations: robot, inspection and storage station. When a raw material is introduced by an operator production orders are delivered, so that each module is aware of their tasks and roles on production. When batches of raw material are deposited onto the belt-conveyor (Conveyor agent), it must be aware at any time of which tasks are designated to each pallet that is navigating on the conveyor, and depending on the assigned task it can stop pallets at different docking stations in order to execute a process. When raw material is stopped at robot docking station, it could be delivered to CNC machine or assembly buffers, these tasks are performed by the manipulator (Motoman UP6). What to do and when has to be done, are examples of the information that order agents deal with the cell, specifically to those executable agents in charge of that area or cell section.

6 Validating reconfigurability and agent implementation

Previous to this section, the ITESM manufacturing cell was described in detail in order to make a global view of how elements are initially set and how original operational flux is performed in a traditional environment. In

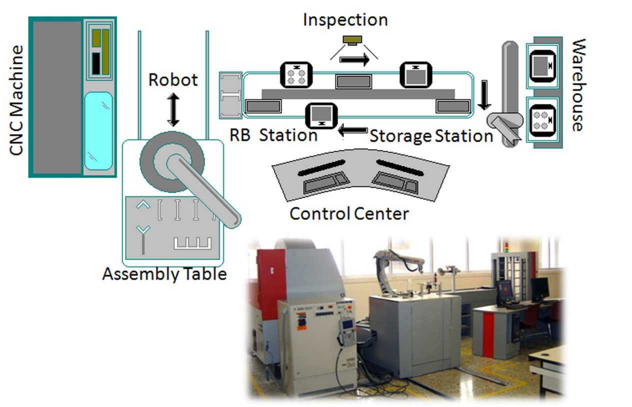


Figure 5: Layout of ITESM Flexible Manufacturing Cell.

this section the configuration of the cell will be altered physically in a non-dramatic way to ensure reconfigurability after implementing the "Holonic-ready" platform, also two elements with no previous interaction will have to cooperate in order to achieve a common goal. It is essential to avoid long setup times, extra physical wiring, or extra monetary investments, to demonstrate the feasibility of migration. The study case begins with some physical changes, figure 5 shows the layout of the ITESM cell, for this application the camera from inspection station will be attached to robot assembly table, its image processing shall construct the perception of robot agent, in other words camera should be the medium that makes the robot able to observe its environment, whereas the robot agents performs decision making process (See fig. 6). The tasks are defined as follows; raw material is delivered by an operator, and this material is formed by a pallet with geometrical figures, as shown in figure 6,

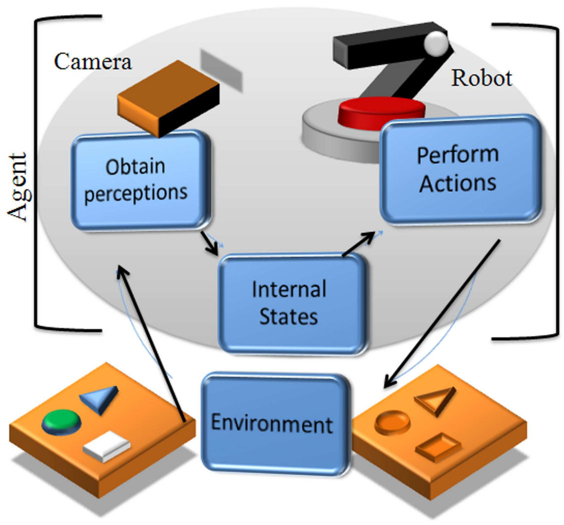
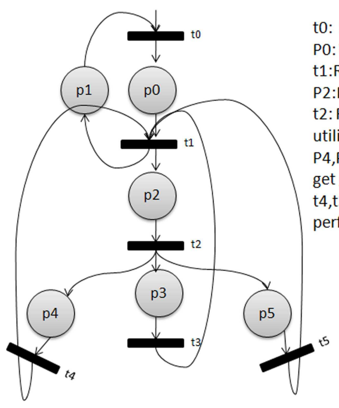


Figure 6: Multiagent abstract architecture for study case.

these figures do not always conserve same patron of placing, thus the robot should perceive by the camera current state from environment, then the robot performs an specific routine or action to deliver each figure to another pallet with a specific location for each geometrical figure (See fig. 7). The petri net demonstrate on figure 7,

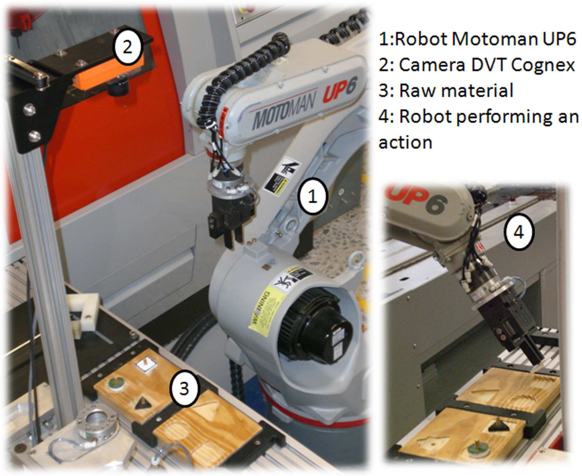


t0: System is activated
P0: Variables are initialized
t1: Raw material is ready
P2: Robot gets ready
t2: Robot has taken a decision based in utilities.
P4,P3,P5: Robots perform actions to get geometrical figures
t4,t3,t5: work is done, ready to perform another action.

Figure 7: A petri net for dinamic study case representation.

how actions and states modify environment from agent perspective, in a dynamic comportment. How often the robot agent performs a determinate route or path is established by the utility each path pays. The amount of utility given for each figure could be assigned by programmer. Nevertheless an agent always tries to maximize the utilities that it can obtain from a task [4], equation 7.

$$Ag_{opt} = argmax_{(Ag \epsilon AG)} \sum_{r \epsilon R(Ag,Env)} u(r)P(r|Ag, Env) \tag{7}$$



1: Robot Motoman UP6
2: Camera DVT Cognex
3: Raw material
4: Robot performing an action

Figure 8: Real working of robot and camera, results.

7 Conclusions and future works

The physical changes were successfully achieved, there are several ways to qualify this characteristic such as time and hardware adaptation, even both aspects were optimized with the usage of the holonic-ready platform, if there were dimension changes on assembly table for example, collision of work space would be also an important problem to solve, nevertheless collision could be avoided adding some extra collision avoidance algorithms, it always will depend on how old-fashioned integrated systems are in the FMC to migrate and its ability to interact. In other words solutions for different elements, will depend on how flexible or communicable they are, as a result we could have several solutions. However preparation of a generic platform that actually could adopt different solutions seems the most urgent issue. The platform shows sufficient flexibility to accomplish the unexpected request of assembling products, as well as showing flexibility in removal, addition and reconfiguration of assembly devices. We could succeed to implement a holonic-ready platform in a generic mode showing its capability for migration to convert common FMSs into RMS agent-based systems. As future work a scheduler of assembly devices shall be developed, and interaction with superior levels such manufacture execution system, both have to be developed in a generic schema to be adopted on the platform, opening different research lines such as logistics and planning for intelligent manufacture, and technological migration.

8 Acknowledgements

This work is a contribution for the Rapid Product realization for Developing Markets Using Emerging Technologies Research Chair, Registration No. CAT077. The authors wish to acknowledge the support of the Tecnologico de Monterrey, Campus Monterrey, Graduated Program department and the Mexican Science and Technology Council (CONACYT) sustaining under Grant J110256.

References

[1] E.J. Lee, A. Toguyeni, N. Dangoumau, “A Petri Net based approach for the Synthesis of Parts’ Controllers for Reconfigurable Manufacturing Systems,” *SICE-ICASE International Joint Conference 2006*, sice, pp. 5567-5572, 2006

[2] Kusiak, Andrew. *Intelligent manufacturing systems* Prentice Hall, c1990

[3] Wooldrige, M. *An Introduction to MultiAgent Systems* 1st Edition, 2002.

[4] Vidal, J.M. *Fundamentals of Multiagent Systems Textbook* 1st Edition, 2002.

[5] Koestler, A.: The Ghost in the machine. Hutchinson, London (1967) Danube edition, with new preface: 1976.

[6] Fletcher, M., McFarlane, D., Thorne, A., Jarvis, D. Lucas, A.: Evaluating a Holonic Packing Cell, First International Conference on industrial applications of holonic and multi-agent systems., HoloMAS, Prague (2003).

[7] Gaxiola, L., Ramrez, M., Jimenez, G., Molina, A.: Proposal of Holonic Manufacturing Execution Systems Based on Web Services Technologies for Mexican SMEs, First International Conference on industrial applications of holonic and multi-agent systems, HoloMas, Prague (2003).

[8] Gaxiola, L.: Holonic environment integration methodology for metalworking SMEs manufacturing systems. MSc. Thesis, ITESM, Monterrey, Mexico (2004).

[9] Jorge, M. Gamboa, Miguel, J. Ramirez, “A Generic multi-agent architecture design in a FMC, implementing distributed intelligence,” *6th International Workshop on Practical Applications of Agents and Multiagent Systems*, Salamanca, Spain, pp. 11-20 2007

[10] Van Leeuwen, E.H., Norrie, D.: Intelligent manufacturing: holons and holorachies. Manufacturing Engineer, 76(2),86-88, (1997).

[11] Saad, K. Kawamura, and G. Biswas, Performance evaluating of contract Net-Based hierarchical scheduling for flexible manufacturing systems.

[12] MESA International, Controls definition and MES to controls data flow possibilities, White Paper No. 3 ed., 1995

[13] T. Arai, Y. Aiyama, M. Sugi, J. Ota.: Holonic assembly system with Plug and Produce, Elsevier(2001)

[14] Weiss, G.*Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence* 2nd Edition, 2000.