

A Holonic Manufacturing System implementation

Nuno Silva

email: nsilva@dei.isep.ipp.pt

URL: <http://www.dei.isep.ipp.pt/~nsilva>

Paulo Sousa

email: psousa@dei.isep.ipp.pt

URL: <http://www.dei.isep.ipp.pt/~psousa>

Carlos Ramos

email: csr@dei.isep.ipp.pt

URL: <http://www.dei.isep.ipp.pt/~csr>

Instituto Superior de Engenharia do Porto (ISEP/IPP)
Departamento de Engenharia Informática
Rua São Tomé, 4200 PORTO, Portugal
Phone: + 351 - 2 - 8340500; Fax: + 351 - 2 - 821159

Abstract

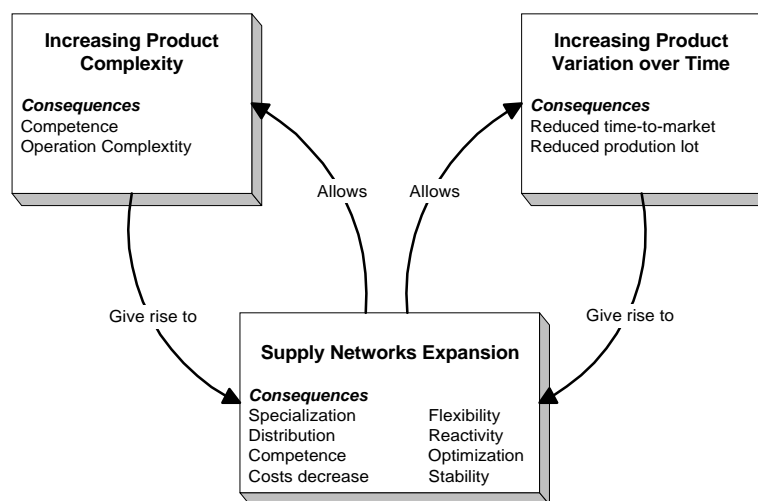
Manufacturing systems must adapt to new challenges impacting the economy. New structures, behaviours and requirements arise in response to those challenges. These elements are systematised and different problem approaches are analysed. Multi-agent concepts generally and holonic concept specially are studied. Based on its requirements and characteristics several frameworks are compared. Besides, a new framework is presented and its characteristics and features are briefly described. The holonic manufacturing system architecture and the implementation itself is presented. Main contribution to the system is the definition of complex resources, and cooperative blackboards. Additionally, protocols and rules for identification, authentication, registration and information publication through the system must be defined. Dynamic scheduling is adopted and a well-known scheduling method should be adjusted to meet the multiple resource for operation.

Keywords: Multi-Agent Systems, Holonic Manufacturing Systems, Holonic Framework, and Scheduling.

1. Introduction

During the last years, three trends are challenging manufacturing systems: (i) increasing product complexity, (ii) increasing product variety on reducing time-to-market and (iii) expansion of supply networks.

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, virtual and federation enterprise.



These new organization concepts lead to new approaches in production and therefore in Manufacturing Systems

Figure 1 - Consequences and causes relation between trends

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 questions the manufacturing system should start to deal with. 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). From the logical point of view we are also dealing with a distributed problem, because several tasks¹ can be carried out at the same time in different resources. Due to these

¹ By task we mean a set of operations to perform resulting in a final item (e.g. make a chess pawn). An operation is a small part of a task, for example in the "make a chess pawn" task, one operation is drilling and another one is painting.

reasons the framework of Distributed Artificial Intelligence (DAI) and Holonic Manufacturing Systems for dynamic scheduling of industrial tasks is proposed [Sousa, 96].

Section 2 describes some Multi-agent and Holonic concepts and comparing them, help us to explain why they were chosen as base paradigms for our project. In section 3, several multi-agent frameworks are analyzed and introduce a new one developed at ISEP-IPP. Section 4 presents the Dynamic Scheduling Manufacturing System Architecture and describes some structural issues and operational behaviors concerning communication issues. Section 5 describes two different methods for scheduling, relating it with the two different organizational and coordination possibilities. In the last section, conclusions about already done work, and some remarks about questions not yet addressed are discussed.

2. Multi-Agent Systems and Holonic Manufacturing Systems

Until few years ago, CIM concept was considered satisfactory enough treating enterprise-manufacturing requirements. However, taking in to account the described set of new organizational and economic concepts, it becomes clear that the centralized CIM approach is not the answer. On the contrary, as stated in [Solberg, 93], these new concepts suggest autonomy, distribution, and flexibility, while stressing the need for coordination among production units.

Therefore, we suggest an approach based on an element called agent. Such manufacturing system is called Agent-based Manufacturing System (AMS). Table 1 summarizes and compares both concepts.

	CIM	AMS
Structure	Static and Centralized	Dynamic and Distributed
Organization	Hierarchic	Heterarchic
Behavior	Predefined	Dynamic
Control	Efficient	Reactive and Reliant
Plans	Predefined	Cooperation-Coordination
Performance	Good and Known	Coordination dependent
Humans	Excluded	Added value
Flexibility	Limited	Nature

Table 1 - CIM vs. AMS characteristics and approaches

An agent is a rational entity sensing and acting on its environment in order to achieve its objectives. There are several properties needed by and found in agent. Among others (quote):

- **Autonomy:** agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.
- **Social ability:** agents interact with other agents (and possibly humans) via some kind of agent-communication language.
- **Reactivity:** agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it.
- **Pro-activeness:** agents do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative [Jennings, 95], [Jennings, 96], [Franklin, 96].

Multi-agent system² refers to society (or community) of agents, meaning there exist some sort of organization, cooperation and coordination among agents to achieve a goal or goals. The Holonic Manufacturing System was proposed by the Intelligent Manufacturing Systems consortium to address the next century challenges in manufacturing systems, thought there are other philosophies based on agents, concerning organization and behavioral characteristics of modern manufacturing systems, like Fractal Factory and Bionic Manufacturing. Table 2 resumes and compares the concepts.

	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

² Not to be confused with **multi-agent** which refers to an agent composed of several different agents.

			through communication
Main Focus	Reactive and Flexible	Multi-dimension approach	Background Concepts & Natural adaptation

Table 2 - Comparison of three concepts for multi-agent manufacturing systems (from [Tharumarajah, 96])

Bionic Manufacturing System (BMS) is the concept most reactive and flexible to change, since its behavior is based on reaction to changes in environment. The Fractal Factory is the one who match better the enterprise modeling, since it adopt a multi-dimension analysis, i.e., the modeling analysis comprises between others, technical, human, organizational and functional parameters. The Holonic paradigm is a multi-agent specialization, in the point of view that all the characteristics observed in agents could also be observed in holons. However it matches the requirements for modern manufacturing systems. Its organization is based on sets of entities that group with each other, forming intermediate stable entities. These entities, as pointed by Koestler when analyzing the Simon watchmaker parable [Koestler, 67], allows a complex system to be efficient and reliant to disturbances, and at the same time flexible and reactive to changes. Entity autonomy is granted and restricted by a common canon, i.e., a set of basic and general cooperation rules. In order to achieve the proposed goal(s), entity must coordinate actions with other entities through conversations, both lateral and vertical. This type of coordination is very efficient since intermediate stable entities has subordination to the upper levels and supra-ordination to their parts [Bongaerts, 96], but at the same time, the entity itself is a holarchy with these characteristics, thus forming a very coherent system.

When structuring a modern manufacturing system, the stressed properties should be essentially, and therefore, the holonic approach seems appropriate and natural.

3. Framework

In a Multi-agent/Holonic architecture, in which elements are characterized by autonomy, social ability, reactivity, coordination, etc., communications and environment inference infrastructures becomes crucial. Therefore, any holon should have a minimum set of features:

- Communication mechanisms:
 - Peer to peer, Group and Routing;
 - Identification, Authentication and Name Services;
 - Security;
- Goals definition:
 - Procedural, Declarative or both;
- KB Policies:
 - Black-Board vs. Broadcast vs. Specific demand;
 - Interaction (Query) language;
 - Security;
- Conversation Policies
 - Conversation templates;
 - Coherence methodologies, mechanisms and tools;

In respect to requirements, some frameworks have been developed over the last years. Some are very complete while others focus on some aspect of requirements. Table 3 presents four of them. Agent Builder is a commercial integrated software toolkit for agent development from Reticular Systems Inc, KAoS is a private product developed by Boeing Information and Support Services on supervision of Jeffrey Bradshaw. JATLite and JAFMAS are academic products and its use is free from charge. JAFMAS was developed by Deepika Chauhan for its doctoral thesis in University of Cincinnati while JATLite is the result of long term research done on agents community at Stanford University.

All the four rely on JAVA concepts and mechanisms to provide communication mechanisms, including both static and mobile agents.

	JAFMAS	JATLite	KAoS	Agent Builder
Group Communication	Yes	Yes	Yes	Yes
Routing Communication	No	Yes	Yes	No
Identification	Yes	Yes	Yes	Yes
Authentication	Yes	Yes	Yes	Yes
Agent Name Services	Yes	Yes	Yes	Yes
Security	Basic	Yes	Advanced	Yes
Black-Board	Subscription	Yes	Application	Application

Goals Definition	Application	Application	Yes	Application
Conversation coherence	Based on COOL	No	Templates, Suites	Several tools
Interaction Language	General Speech-act	KQML (standard)	Meta-primitives	No
Focus	Interaction Coherence	Network Reliability	Very Complete	Development Toolkit

Table 3 - Multi-Agent frameworks comparison

The fundamental characteristics that make KAoS the more complete framework are the use of distributed object technology, the conversation templates and query languages primitives. The first element guaranties security in interaction, while the last two allow a simple, fast but efficient way to develop agent-based system. JAFMAS and JATLite frameworks complement each other in functionality. JAFMAS focus on interaction coherence while JATLite emphasizes network and communication reliability. JATLite consist of a class-layered framework that can be used for different needs depending on the chosen layers. Agent-Builder is a commercial product based on agent templates and "action libraries", developed for non-expert users.

Test cases based on JAFMAS and JATLite were analyzed while KAoS information and use is very restricted. Agent-Builder is still in development thus, more analysis considerations are difficult.

3.1 Developed framework

When this test-case implementation started, July 1996, there were not many known and available for non-commercial use frameworks, so it was decided to develop a framework concerning the requirements.

The developed framework, **HF**W - **H**olonic **F**ramework - attempts to address solutions concerning the requirements presented. At that time, it was clear that it must focus in a sub-set of those requirements:

- Define a basic set of rules to problem decentralization;
- Define and provide "standard" communication mechanisms and rules;
- Hide communications complexity;
- Provides synchronous and asynchronous communications;
- Provides internal knowledge manipulation functions;
- Provide information provider (blackboard) mechanisms and functions;
- Support for multiple protocols;
- Simplify and accelerate development.

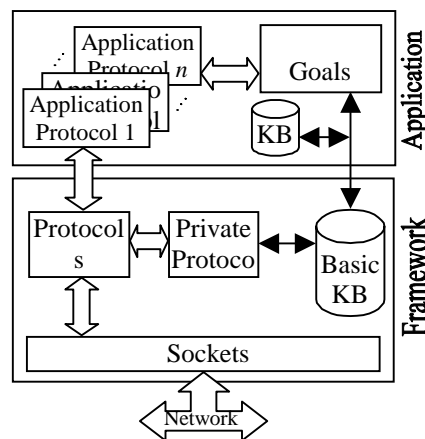


Figure 2 - HFw internal structure: protocol and knowledge base mechanisms

The implemented framework has been developed in Microsoft Visual C++ using the MFC classes and concepts. The main communication characteristics are:

- Implemented using the MFC sockets, both synchronous and asynchronous;
- Peer-to-peer connections;
- Support for messages, both objects-based and raw format.

The knowledge base functionalities include:

- Three types of information: What, Where and How;
- Information Request and Publishing functions. The framework already include a set of functions for data management over the holarchy;
- Every holon is information provider, since it must inform holarchy about is capabilities;
- Requests are always made via specific demand to owner. Any request for information must be directed to a specific holon;

- Every holon is a holarchy blackboard, since it can publish data related and belonging to others.

Protocols mechanisms include:

- Protocol router that is responsible for the routing of messages to conversations. The use of object oriented mechanisms simplify and clarify the process;
- Predefined protocol related to knowledge base request and publishing is already defined and is application transparent;
- Other protocols and conversation rules are application specific;
- A memory area is supplied for each connection. It can be used to maintain related connection state and conversation information.

Goals definition is application specific and no support is supplied. Since the framework itself is a MS-Windows DLL, it is possible to have both procedural and declarative goals definition.

4. Manufacturing system implementation

The architecture presented was first proposed in [Sousa, 96], and is summarized in Figure 3. This figure intends to illustrate part of the Manufacturing System holarchy.

The manufacturing system is composed of several high-level holons (e.g. Process Planning, Scheduling, Stock Management) made up of other holons. At second level, there are the resource, product, and task holons (among others), which are parts of several high-level holons at the same time (e.g. scheduling and process planning holons use resource holons). Thus, forming a highly flexible and reactive structure, making no (or little) compromise with efficiency.

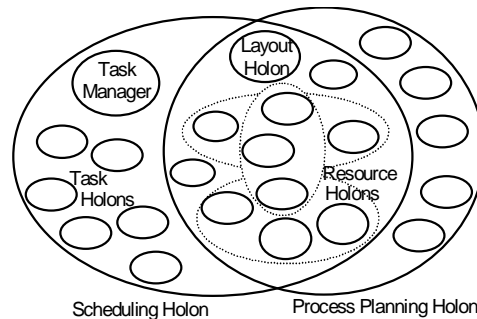


Figure 3 - Dynamic Scheduling manufacturing System Architecture

Clearly this kind of architecture is much more flexible than the static and traditional CIM architectures. However, a great effort must be put in the co-ordination between holons to maintain the overall systems coherence [Sousa, 96], [Sousa, 97].

Since its first proposal, two important features have been added:

- The resources can be grouped in dynamic holarchies (represented by dashed circles in Figure 3), which behavior and existence is dynamically defined by the Layout Holon in conjunction with Process Planning Holon depending on the production requirements over the plant floor. Layout holons are proposed in order to solve bottleneck and unbalancing problems in resources and production lines. Another important function is to coordinate the transport between resources. It is not obvious that a resource can participate in production if the material transport between the previous or the next resource is difficult or even impossible.
- Resource holon structure has been expanded. Essentially, resource can represent a simple undivisible entity or a complex structure that represent multiple internal entities (Figure 4). Examples of this kind of resource are an enterprise inside a virtual enterprise, where the first must be seen as a simple resource, while internally it must be structured like a complex structure of entities. Such holon must be understood as an holarchy with all the functionalities expected. Its main characteristic is the common interface for all the entities inside, i.e., every entity inside has no recognition outside the boards.
- Introduction of the Holonic Name Service (HNS) and Blackboard, since they are not provided by the framework.

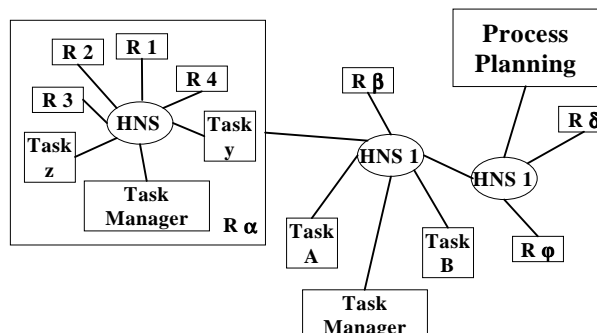


Figure 4 - Different resource internal structures

The case study we are developing is concerned with testing the presented concepts and ideas, checking its validity and utility. Currently, the work is focused in communication, protocols, type and contents of the messages, addressing, inference mechanisms, etc.

The main competencies of the different holons are:

- The task manager holon: although not mandatory, its presence may be useful for performance and functional issues [Sousa, 97]. This holon is responsible for start the scheduling conversation and inter-task deadlock and conflict avoidance; In that sense, it stores a list of the availability of a resource to negotiate;
- Task holon is responsible for the definition of the scheduling negotiation policies, for example the scheduling method and heuristics concerning the time intervals chosen process. The facts and situations that motivate the choice and the inherent responsibilities are different.
- Resource is responsible for the management of its occupation agenda, based on different policies defined by the task holon. If the resource is complex, then the resource constitute itself a manufacturing system holarchy;
- HNS is responsible for the holons identification, authentication and registration in holarchy. The multiple cooperative HNS was been adopted. In case of HNS connection failure, any holon must reconnect again, in order to participate in the holarchy again;
- Blackboard (BB) is the holon responsible for the data, information or knowledge publishing, Multiple cooperative BB structure was adopted.

More advanced features should be presented in order to better understand the holarchy operation, especially concerning with identification, authentication and knowledge management.

- Concerning identification and authentication, the main rule is: any holon register with only one HNS. This is accomplished by the HNS-Holon specific protocol. When the holon born, it broadcast the name requesting a HNS connection. All the HNS holons receiving the message, send a "already defined?" request to its HNS partners. If any HNS respond, it means that the holon is already registered and the request is invalidated. Otherwise, the first HNS in the list establish the connection and the authentication and registration proceed.

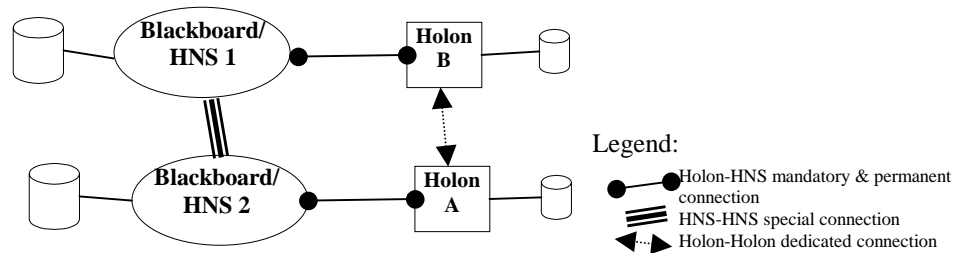


Figure 5 - HNS-BB integration and coherence maintenance

- The most important issue concerning information is the distributed cooperative directory services (blackboards), intended to distribute the information and service for different sites. Distribution allows reliance to disturbance while provoking some classical negative consequences in performance and information coherence. However, the "publishing protocol" specifies seven base rules to guarantee coherence and increase performance:
 1. Any holon connects to only one BB. Normally, but not mandatory, the HNS is also BB (HNS-BB), therefore only one connection from the holon to HNS and BB is established. If not, some regulation mechanisms are needed;
 2. Information belongs to the entity (holon) who created it and only that holon can change or delete it;
 3. Any information requests are addressed to the holon HNS-BB connection or to the information owner;
 4. HNS-BB holons are interconnected by special protocols. When some information is requested that is not available in that HNS-BB, a special request is sent to the others, who must reply with the information. Because the information is not replicated (duplicated), only one or none HNS-BB own that information and thus coherence is maintained. Each HNS-BB, permanently connects to all the others in order to quickly access information existing in the holarchy;
 5. When information is supplied, it has a validity period, after this time it can change in source and thus, becoming unusable.
 6. Special inter HNS-BB mechanisms can be used to guaranty that information requested is valid until any declaration in contrary, i.e.; the requested information is valid until the publisher declares the contrary. This mechanism allows replication while maintaining coherence, since HNS-BB can not perform changes in information replicated from others.

Other protocols used are described in [Sousa, 96]. The implementation and highlights relationships between the different holons are briefly presented in Figure 6 (each arrow represent a different protocol).

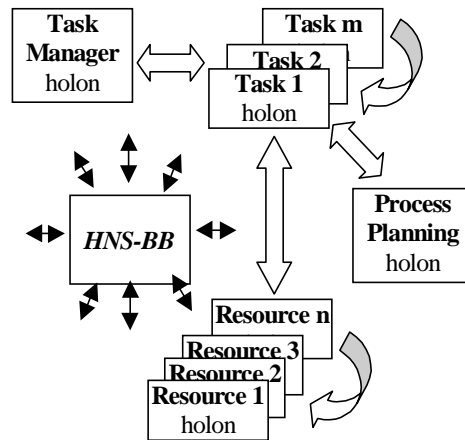
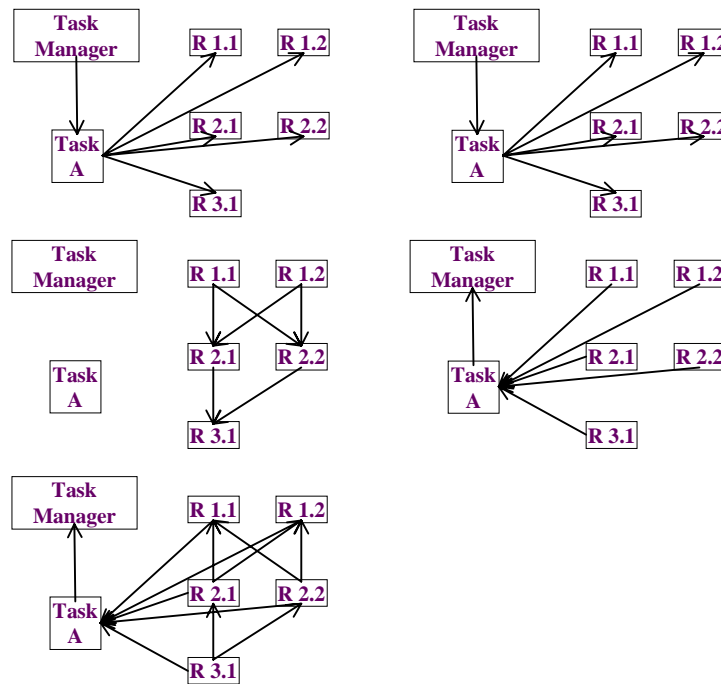


Figure 6 - Distributed Scheduling System protocols case study diagram

5. Dynamic scheduling methods

The holarchy structure was defined based on two assumptions: the first is that each task operation can be executed in several different resources and the second that are two different types of resource:

- Local, cooperative and coordinative resource, in response to a more friendly environment where the resources negotiate with each others for the operation execution, attempting to chose the best for the overall system;
- Non cooperative resource is related to resources competing for the operation execution. Communications limitations, coordination and cultural problems are some questions that lead to this resource position. In this approach, resources do not coordinate with others, they respond to some higher level entity that coordinate the



a) Coordinative resource

b) Non-coordenative resources

Figure 7 -Two different scheduling methods

answers, in this case, the task holon.

Therefore, a different method is applied to each type of resource or situation. The scheduling methods described next are based on the centralized method presented in [Ramos, 95], that for each operation only one resource is available, thus no negotiation is necessary, just time intervals coordination.

Our main scheduling goal is to schedule manufacturing task in different resource considering due dates, though the task can be represented by a time precedence graph, i.e., the operations are performed in a defined order in time.

If the structure is based in coordinative resources, the applied method includes three phases. The forward influence phase, the backward influence phase and decision phase.

Forward influence phase

The forward influence phase starts when the task holon request all the resources for each operation. The earliest operation resources process its agenda to check all the possibilities until the due date. resource influences the next operation (candidates) resources time intervals. It means that the next resource process only the intervals that are compatible with the previous. The process concerns to determine the time intervals possible to execute the operation, considering the operation time span and the remaining operation time span. After each resource process its intervals it passes its own list to the next operation resources candidates. As the intervals flow, there are intervals that are not possible to use and discarded. Like this, the earliest operation resources must have that notification in order to also discard it.

Backward influence phase

In that sense, a backward influence phase is needed. When all the list intervals arrive to the last operation resources the backward influence phase starts. This phase consists in process the final time intervals and sends it to the previous operation resources. Each resource must coordinate the received intervals with its owns and discard the incompatibles. At the same time, each resource must send its final time intervals to the task holon.

Decision phase

When all the resources final time intervals arrive to task holon, the coordination-phase influence starts. The different resources time intervals for the different operations must be coordinate in order to create a list of relations of type "operation-resource-time interval". The coordination process consists in choose such values for each operation that are compatible with the other operations values.

If non-coordenative resources constitute the holarchy, the influence phases are not performed. Time intervals are requested to resources, as in the previous method. The resources, based on information about operation, processes its agenda in order to calculate is availability to execute the requested operation. The result is the resource final time interval list.

The task must start the coordination of the different operation-resource-time intervals. Briefly, the coordination process seems like the influence phases in the previous method, but now, the needed information is present in the same holon.

In the influence phases, the number of messages and the time intervals transferred are very high after a few operations-resources process, thus the combinatorial explosion happens easily. Some work is already done in this subject, but combinatorial explosion is practically unsolvable without applying heuristics. Should the heuristics be applied in negotiation, choosing or both phases?

6. Conclusions

This paper discussed the issue of implementing holonic systems for scheduling of manufacturing tasks.

The motivations and needs for such systems are discussed. The reasons to follow the multi-agent system and specially the holonic approach are presented and compared with other approaches.

Multi-agents systems are very different from traditional systems, thus, the supporting software has special requirements that were listed and analyzed. Based on that list and because multi-agent systems are difficult to build and debug from scratch, it was decided to develop the holonic manufacturing system employing a framework. Hence, four frameworks: JAFMAS, JATLite, KAoS and Agent-Builder, have been analyzed, characterized and compared. The framework used for implementation is HFW, a new framework developed at ISEP/IPP labs. Its characteristics and functionalities are briefly presented.

The proposed holonic system architecture is presented and its main components described. Holons competencies, structure, behavior rules and some operational questions are analyzed, specially concerning communication, coordination and conversations issues.

Finally, two different scheduling approaches are presented, based on two different types of resources and relations between. Consequently, two different scheduling methods are presented and its process described.

There are several questions that need future work. Concerning the framework, some communication issues need some development, e.g. group communication (multi-cast) and relating to the knowledge base, e.g. expand the information types allowed.

Concerning the manufacturing system itself, many topics request hard work, especially concerning to conversations. The development has been done using the timed Petri net approach. About scheduling, the main problems relate to combinatorial explosion in negotiation and time intervals choosing process. The Layout holon logical and functional approach remains dubious.

References

- [Bongaerts, 96] L. Bongaerts, J. Wyns, J. Detand, H. Van Brussel, P. Valckenaers; Identification of Manufacturing Holons; Pre-Proceedings of the European Workshop on Agent Oriented Systems in Manufacturing, Berlin, Germany, 1996.
- [Bradshaw] Jeffrey Bradshaw; KAoS: An Open Agent Architecture Supporting Reuse, Interoperability, and Extensibility; Research and Technology; Boeing Information and Support Services.

- [Franklin, 96] Stan Franklin, Art Graessar; Is it an Agent or Just a Program? : A Taxonomy for Autonomous Agents; Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- [Jennings, 95] Nick Jennings, Michael Wooldridge; Intelligent Agents: Theory and Practice; The Knowledge Engineering Review, vol. 10, No 2, pp. 115-152, June 1995; Cambridge University Press, 1995.
- [Jennings, 96] Nick Jennings, Michael Wooldridge; Software Agents, IEEE Review, January 1996, pp. 17-20.
- [Koestler, 67] A. Koestler; The Ghost in the Machine; Hutchinson & Co, London, 1967.
- [Parunak, 98] H. Van Dyke Parunak; What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC; Industrial Technology Institute; Forthcoming at CIA'98; URL: <http://www.iti.org/~van/cia98.ps>.
- [Ramos, 95] C. Ramos, Ana Almeida and Zita Vale, "Scheduling Manufacturing Tasks considering Due Dates: a new method based on Behaviours and Agendas", Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Melbourne, Australia, 1995
- [Solberg, 93] J. Solberg, R. Kashyap; Research in Intelligent Manufacturing Systems; Proceedings of the IEEE, Vol. 81, No. 1, January 1993.
- [Sousa, 96] P. Sousa, C. Ramos; Scheduling Orders in Manufacturing Systems using a Holonic approach; European Workshop on Agent-Oriented Systems in Manufacturing; Berlin, Germany, September 1996.
- [Sousa, 97] P. Sousa, C. Ramos; A Distributed Architecture and Negotiation Protocol for Scheduling in Manufacturing Systems; ASI'98, Budapest, Hungary, August 1997.
- [Tharumarajah, 96] A. Tharumarajah, A. J. Wells and L. Nemes, Comparison of the bionic, fractal and holonic manufacturing system concepts; International Journal Computer Integrated Manufacturing, 1996, Vol. 9, No. 3, pp.217-226.