A Framework for Distributed Manufacturing Applications

Paulo Leitão[1,2], Francisco Restivo[2,3]

[1] Polytechnic Institute of Bragança, Quinta Sta Apolónia, Apartado 134, 5301-857 Bragança;
[2] Instituto Desenv. e Inovação Tecnológica, Rua do IDIT, Espargo, 4520-102 Sta Maria da Feira
[3] Faculty of Engineering, University of Porto, Rua dos Bragas, 4099 Porto Codex;
pleitao@ipb.pt, fjr@fe.up.pt

Abstract

The new organisational structures used in world wide manufacturing systems require the development of distributed applications, which present solutions to their requirements. The work research in the distributed manufacturing control leads to emergent paradigms, such as Holonic Manufacturing Systems (HMS) and Bionic Manufacturing Systems (BMS), which translates the concepts from social organisations and biological systems to the manufacturing world. This paper present a framework for the development of distributed manufacturing applications, based in an agent-based architecture, which implements some Holonic and Bionic Manufacturing Systems concepts.

Keywords: distributed manufacturing systems, holonic manufacturing, multi-agents, system integration.

1. Introduction

The application of advanced technologies and informational tools by itself does not guarantee the success of control and integration applications. In order to get a high degree of integration and efficiency, it is necessary to match the technologies and tools with models that describe the existing knowledge and functionality in the system and allow the correct understanding of its operation.

In a global and wide market competition, the manufacturing systems present requirements that lead to distributed, self-organised, co-operative and heterogeneous control applications. In this paper it is presented a framework for distributed manufacturing systems based on the Holonic Manufacturing Systems concepts and supported by an agent-based architecture. This approach specifies an architecture for a generic agent and a set of typical types of agents, each one with specific features and functionalities.

2. Layered approach to distributed manufacturing

The globalisation of the markets and the world-wide competition forces the enterprises to implement new technologies and organise themselves using new concepts in order to maintain their competitivity. Today, the manufacturing enterprises can no longer be seen acting stand-alone, forcing them to reconsider how they are organised and to introduce the Virtual Enterprise concept.

2.1 Overview of distributed manufacturing paradigms

The Virtual Enterprise is a paradigm that can be defined as a temporary alliance of enterprises that come together to share skills and resources in order to better respond to business opportunities and whose co-operation is supported by computer networks [Camarinha-Matos et al, 97]. The term Virtual Enterprise is used because in spite of having all the attributes of an enterprise, it would not be a permanent organisation (for example, a joint venture is one type of Virtual Enterprise, where some enterprises group together in order to achieve a particular and common goal). An example of Virtual Enterprise paradigm is the Boeing company that established a virtual co-operation with several companies in order to produce its 777 aeroplanes. Boeing designs, assembles and markets the aircraft, while an international network of suppliers makes the components.

Supply chain management is other typical concepts associated to the distributed manufacturing systems, which deals with the management of materials, information and financial flows in a network, consisting of suppliers, manufacturers, distributors and customers. Roger Blackwell [Harrison, 92], says that "Supply-chain management is all about having the right product in the right place, at the right price, at the right time, and in the right condition". Supply chains span from raw materials, to manufacturing, distribution, transportation, warehousing, and product sales. As the responsibilities are divided into different enterprises, maintaining a continuous control of the production flow gets very complex.

2.2 Multi-Enterprise Model

The key question faces these new enterprise organisational requirements, is how to model the new business organisations, such as virtual enterprises and supply chain. To solve this problem, is proposed a layer approach to distributed manufacturing, applying the fractal concept [Leitão and Restivo, 99].

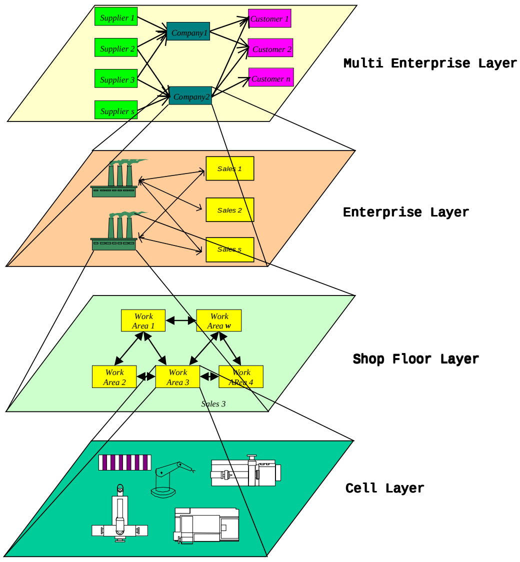

***Figure 1** - The Layer Approach to Distributed Manufacturing*

In the first layer, called Multi-Enterprise layer, it is modelled the interaction between distributed enterprises, acting together in order to achieve a common objective. Each enterprise interacts with their suppliers and customers.

A similar environment is found within each manufacturing enterprise. Zooming into an enterprise shows another distributed manufacturing layer, called the Enterprise layer, where it is possible to find the co-operation between geographically distributed entities, which normally are the sales offices and the production sites. Zooming again into the production sites shows the Shop Floor layer, where the distributed manufacturing control within a production site or shop floor can be found. In this layer, the entities are distributed work areas working together and in co-operation, in order to fulfil all orders allocated to the shop floor, respecting the due dates. Again, zooming into the work area shows the Cell layer, where it is possible to find the co-operation between equipments and humans.

2.3 Requirements

These new ways of business organisation present special requirements and problems, which increase the complexity of the development of control and integration applications. The main requirements that the next generation of manufacturing systems should comprise are [Shen and Norrie, 1999]:

- **Enterprise integration** – in first place the integration of all systems within an enterprise, but also the integration with the systems of other enterprises by networks.
- **Distributed organisational architectures** - distribution of functions, knowledge and operations.
- **Heterogeneous environments** - presence of heterogeneous hardware and software applications.
- **Integration of humans** - integration of humans with software and hardware applications.
- **Co-operation** – co-operation with suppliers, customers and partners.
- **Open and Dynamic structure** - integration of new systems or remission of existing systems without stopping the process.
- **Dynamic Organisation structure**- in order to adapt to the volatility of the global markets.

The problems associated to the implementation of developed control applications to face the previous requirements are [Leitão and Restivo, 99]:

- **Data Translation** - The way to represent and understand the information in a decentralised environment is different for each company. The data exchanged between the companies, over the communication platform, are basically commercial data, using an EDI (Electronic Data Interchange) format, like EDIFACT, and product data, using the STEP protocol or a similar one. However, these standards do not completely solve the data translation problem. Recently, the XML (eXtensible Markup Language) is pointed as the standard for the exchange of data. The XML format allows for tag definition reflecting the structure of the data, which facilitates the data exchange between different sources. The data is converted from the source format to XML in the middle-tier and then transferred to the target entities. The XML allows for designing tailored messages and extensions to the semantics.
- **Decentralised Planning and Scheduling** - The planning and scheduling for a company, working stand alone, is a hard task, due to the high number of constraints involved. However, in a distributed environment, it is necessary to link the planning and scheduling systems of a company with the planning systems of its suppliers. This decentralised planning and scheduling requires the synchronisation of local planning and scheduling systems and a decentralised platform to support the global optimisation.
- **Re-organisation techniques** – the system should present self-organisation features in order to adapt to the external changes. The way to represent the re-organisation techniques and the responsabilities associated to the trigger of the re-organisation is a complex task which require additional research in order to develop a standard model to represent those techniques.

3. Distributed Manufacturing control paradigms

The traditional manufacturing control systems have low capacity to adapt and react to the dynamic changes of its environment, such as the reaction to disturbances and reactions to the market changes. The new paradigms should have the ability to respond promptly and correctly to external changes, and they differ from conventional approaches due to their inherent capability to adapt to changes without external interventions.

In order to achieve the above mentioned characteristics and behaviours, several theories for the next generation of manufacturing systems were proposed, presenting similar concepts and characteristics but with different origins: mathematics for the fractal factory [Warnecke, 93], nature for bionic manufacturing systems [Okino, 93] and social organisation for holonic manufacturing systems [Bongaerts, 98].

3.1 Holonic Manufacturing Systems

The Holonic Manufacturing System translates the concepts that Koestler [Koestler, 67] developed for living organisms and social organisations into a set of appropriate concepts for manufacturing industries. Koestler used the word holon to describe a basic unit of organisation in living organisms and social organisations, based on Herbert Simon theories and on is observations [Bongaerts, 98]. Simon observed that complex systems are hierarchical systems formed by intermediate stable forms, which do not exist as auto-sufficient and non-interactive elements but, on the contrary, they are simultaneously a part and a whole. The word holon is the representation of this hybrid nature, is a combination of the Greek word *holos*, which means *whole*, and the suffix *on*, which means *particle* [Bongaerts, 98].

A holon is an autonomous and co-operative entity of a manufacturing system, which include operational features, skills and knowledge, and individual goals. It can represent a physical or logical activity, such as a robot, a machine, an order, a Flexible Manufacturing System, or even a human operator [Leitão and Restivo, 99].

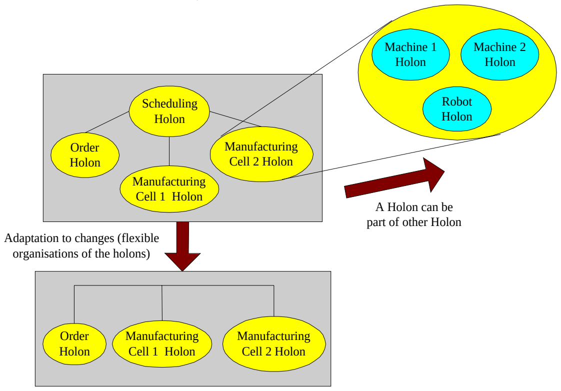

Figure 2 *– Holonic features: adaptation and holon expansion*

The holon has information about itself and the environment, containing an information processing part and often a physical processing part. In the previous figure, the *scheduler holon* only has information part and the *machine 1 holon* has additionally the physical part.

An important feature of HMS is that a holon can be part of another holon, e.g., a holon can be broken into several others holons, which in turn can be broken into further holons, which allows the reduction of the problem complexity.

3.2 Bionic Manufacturing Systems

The Bionic Manufacturing Systems (BMS) have developed under the biology ideas and concepts, and assume that the manufacturing companies can be built upon open, autonomous, co-operative and adaptative entities, which can evolve. The BMS translates to manufacturing systems the structure and organisation behaviour of the living beings, defining a parallelism between biologic systems and manufacturing systems. The cell, organ or living being is modelled in BMS by the *modelon* concept, which is composed by others *modelons*, forming a hierarchical structure. Each *modelon* has a set of static properties and behaviours, which can be combined with others, forming distinct entities also designated *modelons*. The notion of DNA inheritance is translated to manufacturing context by the properties and behaviours that are passed intrinsically to developed *modelons* [Tharumarajah et al., 96]. The biological concept *enzymes* and its role in the living beings is modelled in manufacturing systems by entities called supervisors, which are responsible for the regulation and control of the system. Furthermore, the supervisors also play an organisational and structural role in the co-operation process within the BMS, influencing the relations between *modelons*, imposing self-division or aggregation, in order to adapt and react to the requirements imposed by the environment [Sousa et al., 99].

3.3 Fractal Factory

The Fractal Factory is an open system, which consists of independent self-similar units, the fractals, and it's a vital organism due to its dynamic organisational structure. The fractal manufacturing uses the ideas of mathematical chaos: the companies could be composed by small components or fractal objects, which have the capacity to react and adapt quickly to the new environment changes. A fractal object has the following features:

- ***self-organised***, which means that doesn't need external intervention to reorganise itself.
- ***self-similar***, which means that one object in a fractal company is similar to other object. In other words, self-similar means that each object contains a set of similar components and shares a set of objectives and visions.
- ***self-optimised***, which means that continuously increase its performance.

The explosion of fractal objects into other fractal objects, has the particularity of generating objects which possess organisational structure and objectives similar to the original ones. For [Warneke, 93] the factory of the future will present different dynamic organisational structure, adopting the project orientation organisation in contrast with the traditional function oriented organisation. This approach implies the organisational structure will encapsulate the process and the technology, therefore forming a cybernetic structure.

3.4 Multi-agent systems

One approach, which derives from the Distributed Artificial Intelligence, is the multi-agent system concept. The multi-agent systems can be defined as a set of nodes, designated by agents, that represent the objects of the system [Ferber, 99]. There is not a unique definition for the term agent. However, it is possible to define an agent as a component of software and/or hardware that is capable of acting in order to accomplish tasks. In the manufacturing systems domain, an agent is a software object, that represents manufacturing system objects, such as resources and tasks. The agents are autonomous and intelligent, and they can communicate together in order to perform the required tasks. The multi-agent systems are suitable to the distributed manufacturing environment, since the manufacturing applications presents characteristics like modular, decentralised, changeable, ill-structured and complex, for what the agents are best suited to solve [Parunak, 98]. The multi-agent systems approach is a suitable solution to implement holonic and bionic manufacturing systems concepts.

3.5 Comparation of concepts

The concepts of these paradigms are unified in proposing distributed, autonomous and adaptative manufacturing systems. The concepts of each paradigm differ in their approach to design of these features. The bionic manufacturing and specially the holonic manufacturing fit well into some existing methodologies, such as object-oriented. Instead, the definitive approaches may not be suited for fractals that require a dynamic and multi-dimensional approach (for a deeply study see [Tharumarajah et al, 96]).

4. Agent-based architecture for distributed manufacturing applications

The architecture proposed for the development of distributed manufacturing applications is based in the Holonic Manufacturing Systems and Bionic Manufacturing Systems concepts and uses an agent-based approach to implement those concepts. It is a hard and complex task to define a generic architecture able to model all kind of distributed manufacturing applications, due to the heterogeneity of the distributed manufacturing applications.

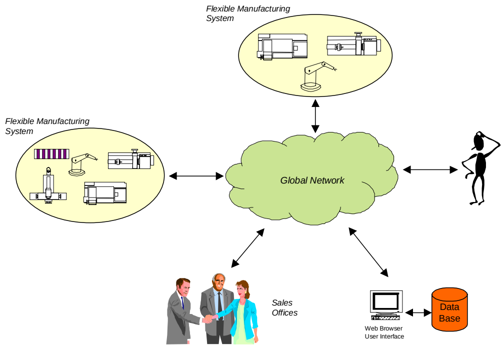

Figure 3 *- Agent-based architecture*

The proposed agent-based architecture has the following main features:

- The system is a set of agents that implement the basic ideas of HMS (such as possibility to represent a human, allow different organisational structures, an agent can be part of other agent, etc) and some ideas from BMS (the role of supervision, etc).
- Each agent is autonomous, intelligent and co-operative.
- The agents can organise itself in different organisational structures (self-organisation), in order to optimise the individual and community objectives.
- The developed agents should belong to one of the agent type class allowed in the architecture.

The architecture presented consider the following classes of agents:

- Supervisors agents, which represents agents that have the tasks and should carry out the supervision of a set of agents. For example, the scheduling and planning of orders are examples of this type of agents.
- Operational agents, which perform the tasks, such as robots, machines, etc.
- Monitoring agents, which acquire information and take an action when something happens (verifies the execution of the actions against the plan).
- Order agents, which have the order to be executed.
- Interface agents, which realises the interface between the system and the user. Additionally, they can provide assistance to the humans during the interaction with the system.

The implementation of this agent-based architecture uses new and powerful technologies, such as JAVA and CORBA environments, network capabilities, such Internet and web-based interfaces, communication standards, such as KQML, etc.

5. Architecture for a generic agent

As we see in the previous chapter, the architecture proposed within a framework for the development of distributed manufacturing applications are based in a set of autonomous and co-operative agents. The next step in the framework design is to specify an architecture for a generic agent that will belong to the system architecture. The next figure represents a modular architecture for a generic agent, based in six main modules and a knowledge database, that contains all relevant (local and global) information to the behaviour of agent.

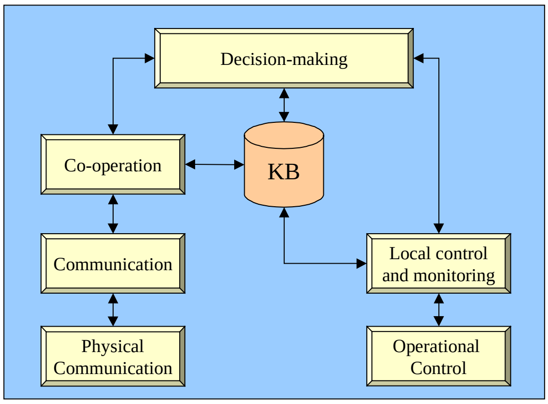

Figure 4 *- Architecture for a generic agent*

All these modules interact itself, and two of them interacts with the external environment. The Physical communication module allows the interaction with other agents, for example to exchange data or to receive orders. The Operational control module allows the interaction with the process, i.e. the execution of the allocated tasks.

In the next figure is represented the functional model for each agent.

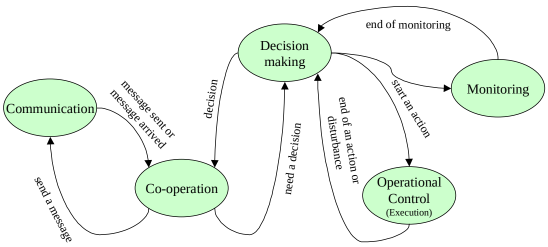

Figure 5 *- Functional model for a generic agent*

In the next point will be described more deeply each of these modules.

5.1 Decision-making Module

In the essence, each agent is autonomous, having autonomy to take decisions about its behaviour. Each agent has objectives to reach: individual and community objectives. The decision-making module is the brain of the agent and controls all activities and information flows. The Decision-making module involves the problem solving and decision-making for scheduling, planning, order allocation and other problems.

In the decision-making process, the module uses the knowledge and information stored in the knowledge data base, and when the available information is not enough, it is started a co-operation process with other agents trying to find out the necessary information to take the decision.

Additionally, the agent is intelligent and the decisions can be taken based in the knowledge acquired in previous experiences, during a learning process from experience.

5.2 Co-operation

In a distributed environment, the co-operation assumes an important key in order to achieve the system goals. There are three types of co-operation:

- Physical co-operation, when two or more agents synchronise its physical activities, in order to achieve a common goal, for example the load of a machine by a robot.
- Exchange of information, when an agent needs additional information to take a decision, for example determine if should accept the task or not.
- Negotiation in task assignment, when during a task allocation process, it is necessary to find out the best way to allocate the tasks to resources, for example in the scheduling of transportation tasks through several AGV's.

The objectives of this module are:

- Establish the mechanisms to implement the communication between distributed agents.
- Guarantee the right communication with other entities, through a protocol.

This module defines a negotiation protocol based on the well-known Contract Net Protocol (CNP) [Smith, 80] and mechanisms to support the co-operation. Based in the demands from the decision-making modules, asking for a specific co-operation, and in the information stored in the KB (target agents, etc), this module applies the pre-defined mechanisms to start the co-operation (makes the management of the co-operation process).

5.3 Communication Module

The communication module deals with the need to standardise the interaction between distributed agents. Normally, the co-operation requires the communication, but there are cases when the co-operation don't imply communication: example when the agents have the pre-defined objectives, and controls its behaviour in order to achieve those objectives, co-operating but without necessity to communicate.

In this architecture the definition of the communication module is only focused for the cases of co-operation which requires communication. In this way, this module defines the type of communication (communication connection, medium and intention) and the communication language, using the KQML (Knowledge Query and Manipulation Language) language. The KQML is an evolving standard agent communication language and protocol, developed as part of the DARPA initiative, for exchanging information and knowledge, which is aimed to develop techniques and methodology for building large-scale knowledge bases which are sharable and reusable [Ferber, 99]. KQML is both a message format and a message-handling protocol to support run-time knowledge exchange among agents, but is indifferent to the format of the information itself. KQML focuses on an extensible set of performatives, which defines the allowed operations (message types) that agents may attempt to perform during the knowledge sharing process. In addition, KQML provides a basic architecture for knowledge sharing through a special class of agent called communication facilitators which co-ordinate the interactions of other agents [Nwana and Woldrige, 96].

An additional aspect in the communication module is to define standard ontologies that defines the vocabulary (i.e. the terms) that will be used in the communication between agents, and the knowledge relating to these terms. The purpose of ontologies is to create shared understanding between co-operative, enabling the exchange of knowledge among agents and the capability to reuse that knowledge [Gruber, 93]. There are several defined ontologies, but the well known is KIF (Knowledge Interchange Format) [KIF].

5.4 Local Control and Monitoring Module

This module intents to control and to monitor the operational execution of the agent. In the control part, the dispatch and management of actions are done, while in the monitoring part, there are two different types of monitoring: active and passive.

Passive monitoring is the type of monitoring that does not involve any alarm event driven mechanism. The request for passive monitoring comes from the decision-making unit, which wants to know some specific information, such as the current status of an order or the capacity of a production site. The active monitoring is related to the alarms and event driven systems concerning to parameters of decision-making units. These alarms can be both disturbances in the resources and manufacturing site, which became not able to execute the orders allocated, and delays in the planned production and delivery dates. In order to fulfil this requirement, the active monitoring sub-process implements an event driven system, based in notification of occurrence of alarms subscribed by local decision-making units [Tönshoff et al, 2000].

5.5 Physical Communication and Operational Control Modules

The physical communication and operational control modules belong to the physical layer, which will be the platform for the distributed applications, and are responsible for the iteration with the external environment. The physical communication allows the iteration with other agents and uses CORBA or DCOM standards. The operational control interacts with the environment allowing the execution of tasks assigned to the local agent.

The development and implementation of the presented architecture will use several technologies and tools*:*

- A programming language or developing environment (JAVA).
- Communication protocols for distributed applications (DCOM or CORBA).
- A protocol for the implementation of data exchange between distributed entities (XML).
- The Web platform and technologies to support the easily interaction between distributed entities.
- A library of functions that enable the building of distributed control applications.

6. Knowledge Base module

The information stored in the Knowledge base involves:

- Several types of knowledge, such as constraints, objectives, procedures, rules and experience.
- Organisational structures and techniques.

The approach to the Knowledge base (KB) has two dimensions: individual-community information and generic-specific information. The individual information is related to the local data of the agent, such as local objectives, while the community data is related to the data concerning the system as a whole, such as system objectives.

The generic part of the data is concerned to the common information to all agents and the specific part of data is related to the specific information for each agent. This data is concerned to all information about the tasks, resources, etc (or in a generic way, the objects, the actors, the processes, etc).

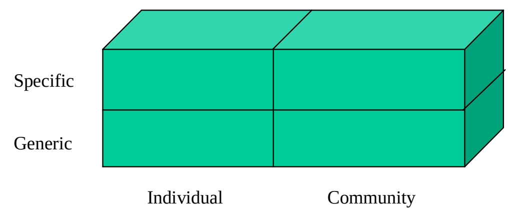

***Figure 6** - Knowledge Base dimensional approach*

In order to standardise the data management, it is necessary to define generic methods to represent the information and the knowledge, such as the rules and constraints.

7. Conclusions

In the decentralised manufacturing environment, such as multi-site, supply-chain and virtual enterprises cases, the development of distributed manufacturing applications assumes a critical issue that leads to the competitiveness of an enterprise. Those applications should present several requirements, such as co-operation, extensibility, integration of humans, and self-organisation features.

This paper presents an agent-based approach to the development of distributed manufacturing applications, based in the Holonic and Bionic Manufacturing Systems. It is presented a generic architecture for an agent and described each component of the architecture.

8. References

[Bongaerts, 98] Bongaerts, L., *"Integration of Scheduling and Control in Holonic Manufacturing Systems"*, PhD thesis, Katholieke Universiteit Leuven, 1998, Belgium.

[Camarinha-Matos et al, 97] Camarinha-Matos, L., Carelli, R., Pellicer, J., Martín, M., *"Towards to the Virtual Enterprise Food Industry"*, in Proceedings of International Conference on Integrated and Sustainable Industrial Production, Edited by Luís Camarinha-Matos, Chapman & Hall, Lisboa, Maio de 1997, pp 73-84.

[Ferber, 99] Ferber, J., *"Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence"*, Addison-Wesley, 1999.

[Gruber, 93] Gruber, T.,"*A Translation Approach to Portable Ontologies*", Knowledge Acquisition, 5 (2), 1993, pp199-220.

[Harrison, 92] Harrison, A., *"Just in Time Manufacturing, in Perspective"*, Prentice-Hall, 1992

[KIF] http://logic.stanford.edu/kif/dpans.html, in March 20th 2000.

[Koestler, 67] Koestler, A., *"The Ghost in the Machine"*. Hutchinson & Co, London, 1967.

[Leitão and Restivo, 99] Leitão, P. and Restivo, F., *"A Layered Approach to Distributed Manufacturing"*, in proceedings of 1999 Advanced Summer Institute - LIFE Cycle Approaches to Production Systems: Management, Control, Supervision, Leuven, Belgium, 21-23 September 1999.

[Nwana and Woldrige, 96] Nwana, H. and Wooldrige, M., "*Software Agent Technologies*", British Telecommunications Technology Journal, 14 (4), October 1996.

[Okino, 93] Okino, N., *"Bionic Manufacturing System"*, In J. Peklenik (ed.) CIRP, Flexible Manufacturing Systems Past-Present-Future, 1993, pp. 73-95.

[Parunak, 98] Parunak, H. V. D., "*What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC*"; Industrial Technology Institute; CIA'98.

[Shen and Norrie, 99] Shen, W. and Norrie, D.H. (1999), *"Agent-based systems for intelligent manufacturing: A state-of-the-art Survey"*, Knowledge and information systems: an International Journal, Vol1, nº 2, pp 129-156.

[Smith, 80] Smith**,** R.G. **"***The Contract Net Protocol: High-Level Communication and Control in a Distributed Solver"*, IEEE Transactions on Computers, Vol. C-29, Nº12, pp 1104-1113, 1980.

[Sousa et al., 99] Sousa, P., Silva, N., Heikkila, T., Kollingbaum, M., Valckenaers, P., *"Aspects of co-operation in Distributed Manufacturing Systems"*, in Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems (IMS-Europe'99), Leuven, Belgium, 22-24 September 1999, pp. 695-717.

[Tharumarajah et al., 96] Tharumarajah, A., Wells, A., Nemes, L., *"Comparison of the Bionic, fractal and holonic manufacturing systems concepts"*, in International Journal of Computer Integrated Manufacturing, 1996, Vol.9, nº3, pp 217-226.

[Tönshoff et al, 2000] Tönshoff, H.-K., Seilonen, I., Teunis, G. and Leitão, P., *"A Mediator-based approach for decentralised production planning, scheduling and monitoring"*, to appear in Proceedings of CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, 21-23 June 2000, Capri, Naples, Italy.

[Warneke, 93] Warneke, H.J.,*"The Fractal Company"*, Springer-Verlag 1993.