A Multi-Agent Based Cell Controller

Paulo Leitão
Polytechnic Institute of Bragança
Quinta S^{ta} Apolónia, Apartado 134,
5301-857 Bragança, Portugal
pleitao@ipb.pt

Francisco Restivo
Faculty of Engineering-University of Porto
Rua Dr. Roberto Frias, P-4200-465
Porto, Portugal
fjr@fe.up.pt

Goran Putnik
Department of Production and Systems Engineering
School of Engineering, University of Minho, 4800-058 Guimarães, Portugal
putnikgd@dps.uminho.pt

Abstract: **This paper discusses the opportunity to use multi-agents technology in automation and distributed manufacturing systems and the expected improvements. To support the discussion, it is described a manufacturing cell control application developed using the multi-agent technology, and the results are compared with other control application developed in the past by some of the authors, using a traditional approach, for the same flexible manufacturing cell.**

Keywords: **Automation Systems, Shop Floor Control, Multi-Agents and Holonic Manufacturing Systems.**

I. INTRODUCTION

The new trends in the market and business environment, like the mass customisation, products short life cycle and e-business technologies, impose a need for agile, re-useable, distributed and cooperative automation systems that improve the competitiveness of an enterprise. The new requirements presented by the new control systems are mainly the distribution and decentralisation of entities, knowledge and skills, the cooperation between distributed entities, the extensibility of entities and components, the transparent communication mechanisms for the easy integration and cooperation of distributed entities, the agile adaptation in the control and in the reaction to disturbances (failures or organisational changes) and interaction with physical devices.

Several approaches were developed to solve the new requirements, and the application of multi-agents technology to automation and manufacturing systems seems to fit well the requirements, mainly the autonomy, the cooperation, the reactivity and pro-activity.

The introduction of multi-agents technology was already presented in previous research projects, such as HOLOS/MASSIVE [1], MetaMorph [2], AARIA [3], Agent-Based Manufacturing Enterprise Infrastructure [4] and MASCADA [5].

The authors intend to present an agent-based manufacturing cell controller implementation, using the architecture ADACOR (Adaptive and Cooperative Control Architecture for Distributed Manufacturing Systems), developed by some of them for the control of distributed manufacturing systems. Using the results of a previous implementation for the same case study, but using traditional approaches for the control, it is possible to compare both approaches and to extract the benefits of the use of multi-agents technology in automation and manufacturing systems.

II. CASE STUDY DESCRIPTION

The flexible manufacturing cell of IDIT's CIM platform, which has already been used in the past to implement a cell controller based in traditional approaches, is the platform used to implement the control application based in the ADACOR architecture.

The shop floor of the platform is organised as a set of four cells: Material Storage and Transportation Cell, Palletising and Calibration Cell, Assembly Cell and Flexible Manufacturing Cell [6]. In the case study context the cell concept refers a group of resources, which are grouped due to its functionality and location.

The palletising and calibration cell is responsible for the assembly of tools for the CNC machines, the calibration of tools and grippers and the palletising and depalletising of the materials that circulate in the shop floor. This cell has a calibrate machine AR2000GA from Elbo Controlli and a palletising table.

The material storage and transportation cell is responsible for the transportation of materials within the shop floor and for the temporary storage of materials (could be raw materials, semi-finished products or final products). This cell has an AGV EFAGV-200-2R-B from Efacec, an Automatic Storage/Retrievial System (AS/RS) from Efacec, and several transfer tables among the shop floor.

The assembly cell has the objective to assembly the components to achieve the final product. This cell has a four-axis SCARA robot Adept Three from Adept Technology. Coupled to the robot, exist a CCD camera from PULNIX, associated to the artificial vision system Cognex 4200EX from Cognex Corporation.

0-7803-7241-7/01/$10.00 (c)2001 IEEE

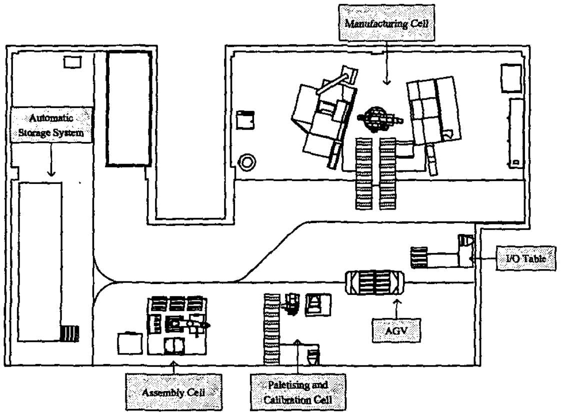
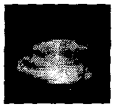
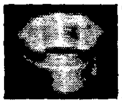


Figure 1 – Shop Floor Layout

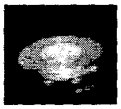
The flexible manufacturing cell has two CNC machines and an anthropomorphic robot for the load/unload of the machines. One of these machines is a turning center *Lealde TCN10*, with a SIEMENS *Sinumerik 880T* controller; the other machine is a milling center *Kondia B500* model, with a FANUC *16MA* numerical control. The robot is a KUKA *IR163/30.1* with a SIEMENS *RC3051* controller. For security reasons this robot is equipped with a SHUNK OPS (*Overload Protection System*). The manufacturing cell has two transfer tables for the containers loading and unloading. These containers bring the material to be operated into the cell and take away the pieces produced.



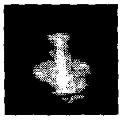
OPERATION SHEET		**Product Name:**	**BASE**
Material:	Stell 64 ϕ 100		
N.°	***Operation***	***Machine***	***Time***
020	Turning–side1	Lathe	60
030	Turning–side2	Lathe	80



OPERATION SHEET		**Product Name:**	**CORPO**
Material:	Stell 64 ϕ 100		
N.°	**Operation**	**Machine**	**Time**
050	Turning–side1	Lathe	60
060	Turning–side2	Lathe	80
070	"Flower" execution	Milling	180



OPERATION SHEET		**Product Name:**	**TAMPA**
Material:	Stell 64 ϕ 100		
N.°	**Operation**	**Machine**	**Time**
110	Turning- side1	Lathe	60
120	Turning- side2	Lathe	80
070	"Flower" execution	Milling	*180*
100	Drilling	Miling	150



OPERATION SHEET		**Product Name:**	**PEGA**
Material:	Stell 64 ϕ 40		
N.°	**Operation**	**Machine**	**Time**
080	Turning–side1	Lathe	60
090	Turning–side2	Lathe	80
100	Drilling	Milling	*150*

Figure 2 – Available Products

All the machines have MAP interface boards for the connection to the control system platform. These interfaces are: **CP 1476 MAP** for Siemens Sinumerik 880T machine controller, **CP 1475 MAP** for Siemens Sirotec robot controller and **GE FANUC OSI-Ethernet Interface** for GE Fanuc 16MA numerical controller.

To test this control application four different products were designed, named *Base*, *Corpo*, *Tampa* and *Pega,* which assembled create a final product named *Cinzeiro*. A simplified operation sheet for each product is represented in the figure 2. The specification of each product involves, among other tasks, the definition of material ID and a list of operations, each one characterised by the description of the operation, the machine type needed to perform the operation and the estimated execution time (in seconds).

III. TRADITIONAL CELL CONTROLLER APPROACH

The traditional Manufacturing Cell Controller, developed and implemented for the Flexible Manufacturing Cell, uses a modified hierarchical architecture approach [6].

The Cell Controller architecture is a set of several modules, whose "brain" is the Manager Module, which is responsible for the control and the supervision of the production process of the manufacturing cell and also for the management of cell resources.

Each physical device has an module, designated by Device Controller, which is customised to the industrial machine, such as production or handling equipment, and it has the responsibility for the local control of the machine, and for the execution of the jobs requested by the high level module.

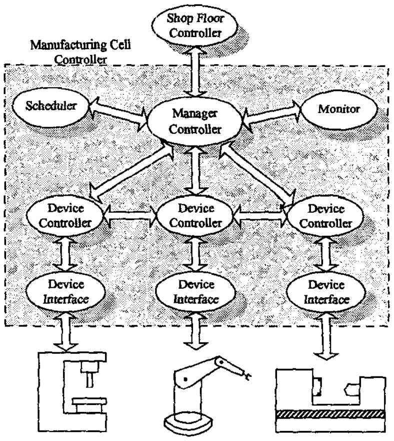


Figure 3 – Manufacturing Cell Controller using a Traditional Approach

The interface between the Cell Controller and each of the industrial machines is implemented using the MMS (Manufacturing Message Specification) communication protocol. MMS is the international standard ISO 9506 [7], and define a standardised message system for exchanging real-time data and supervisory control information between networked devices and/or computer applications in such a manner that it is independent from the application function to be performed and from the developer of the device or application.

Problems Associated

The traditional approach presents the following main problems:

- **Reconfiguration**: fit very well for applications that present a rigid organisational structure. However, it falls down when it is necessary to change the organisational structure (for example, new shop floor layout, new strategies for the hierarchy, etc).
- **Learning and disturbance management**: it is hard and complex to introduce intelligence in the application, in order to optimise its execution and to manage the disturbances and warnings.
- **Distribution and decentralisation**: Doesn't support efficiently the distribution and decentralisation of functions and entities.
- **Code re-usability**: the development of this type of applications based in this traditional approach has the advantage of its simplicity, when compared with other advanced approaches, but the code developed cannot be re-used.

IV. THE AGENT-BASED APPROACH

The main and unsolved problems presented by traditional control approaches are the distribution of functions, cooperation between distributed entities, reaction to disturbances, adaptation to new environments and code re-usability.

The multi-agent systems are defined as sets of agents, which represent the objects of the systems and through co-operation mechanisms perform complex tasks [8, 9]. In the automation and manufacturing domain, an agent is a software object that represents automation and manufacturing system objects, such as tasks, CNC machines, robots, AGVs, transfer tables, PLC devices and sensors.

The multi-agent technology is suitable to the distributed manufacturing environment, since the automation and manufacturing applications characteristics like modular, decentralised, changeable, ill-structured and complex, are best suited for agents to solve [10]. Analysing the benefits of multi-agent technology it is possible to conclude that they overcome problems presented by traditional approaches: autonomy (an agent can operate without the direct intervention of external entities, and has some kind of control over their behaviour), cooperation (the agents interact with other agents, in order to achieve a common goal), reactivity (the agents perceive their environment and response quickly to changes that occur on it), proactivity (the agents do not simply act in response to their environment, but are able to taking the initiative, controlling its behaviour) and adaptation and decentralisation (the agents can be organised in a decentralised structure, and easily can reorganised into different organisational structures).

The expected improvements of the use of multi-agents technology in automation and manufacturing systems are the fast adaptation to system reconfiguration (for example addition or removal of resources, different organisational structures, etc.), re-use of code for other control applications, increase of flexibility and adaptation of the control application and more optimised and modular software development.

V. ADACOR ARCHITECTURE FOR MANUFACTURING CONTROL

The proposed architecture, designated by ADACOR (Adaptive and Cooperative Control Architecture for Distributed Manufacturing Systems), is based on a set of autonomous, intelligent and co-operative agents, forming a multi-agent platform, and implements some concepts derived from holonic (such as the possibility to represent a human and allow different organisational structures) and bionic (such as the role of supervision and the dynamic evolution of the system) manufacturing systems [11]. The agents can organise themselves in different organisational structures, in order to optimise the individual and community objectives, and combine the robustness and the reaction to the disturbances.

The architecture defines a set of agent classes: Operational, Supervisor, Product, Task and System Management agents. The Operational agent handles the interactions with the physical resources, such as CNC machines, robots, humans, PLC devices and sensors. The operational agent has two components: a logic part that control and interacts with the physical device. Additionally, the operational agent allows the interaction with legacy systems.

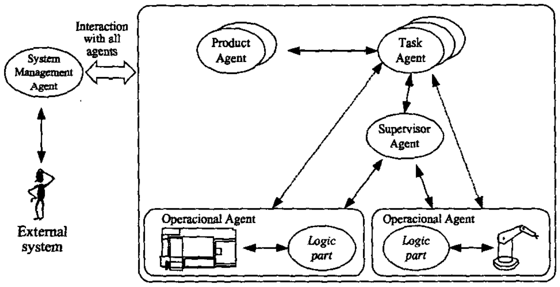


Figure 4 – Agent Classes in the Architecture

The Supervisor agent coordinates and supervises several operational agents, in accordance with the particular organisational structure, and can represent cell controllers.

The Product agent represents the available products in the system and takes care of the product knowledge and the associated information, such as the process plan.

The Task agent is launched to represent the execution of a task in order to produce a product, and it contains the dynamic information about the manufacturing orders.

The System Management agent allows the administration of the system, and the supervision and registration of the agents belonging to the system.

A. *Generic Architecture of an ADACOR Agent*

The architecture for a generic agent that belongs to the proposed architecture is based in three modules and a local database, which contains all relevant information about the behaviour of the agent [11].

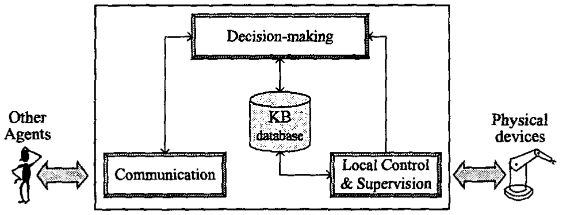


Figure 5 - Architecture for a Generic ADACOR Agent

The **decision-making module** controls all activities of the agent and includes a decision engine, cooperation mechanisms, dynamic organisational techniques and learning mechanisms, in order to take decisions and solve problems. To support the decision-making, the module uses the knowledge stored in the local data base, and when the available information is not enough to make a decision, it starts a co-operation process with other agents to try to find out the necessary information to make the decision.

The cooperation processes manage the cooperation with the external agents and use mechanisms for three different types of cooperation: negotiation (for product specification and order allocation), physical synchronisation and monitoring (passive and active forms).

The **communication module** deals with the need of interaction between distributed agents, using the communication mechanisms defined in previous point.

The **local control and supervision module** intends to control and to supervise the operational execution of the agent. An important feature of this module is the interaction with the physical devices, which is supported by a platform based in CORBA objects that defines the basic services for the control and supervision of the physical devices: write and read variables, and programs manipulation (download, start, stop, etc).

The **local database** stores all knowledge about the behaviour of the agent and the community where the agent belongs. The information stored in the local database involves several types of knowledge, such as objectives, constraints, experience, decision rules, procedures, dynamic information and organisational related information.

B. *Dynamic Structure Re-Configuration*

This approach supports the agile adaptation to different organisational structures and to self-organisation, which allows the re-organisation of the agents into different organisational structures. Special attention is given to a completely heterarchical architecture, and to more hierarchical ones (like the federation approach). However, it is possible to add dynamically new organisational structures.

To support this feature, are used methods to represent the relationships between the agents for each organisation structures and techniques to handle this re-organisation. For example, if one agent leaves the system, it is necessary that all other associated agents update automatically and dynamically their relationships.

In the manufacturing environments there are disturbances (alarms, layout changes, etc.) that deviate the process from the original plans. In case of disturbances the control system should respond dynamically and quickly, using mechanisms that comprise a disturbance engine, which finds out the best plan to handle the disturbance, based in pre-defined rules and in the knowledge acquired with the operation learning.

C. *Communication Mechanisms*

The interaction between the agents in the architecture is asynchronous and uses the TCP/IP protocol. The communication language will be FIPA (Foundation for Intelligent Physical Agents) and the format of messages for data translation uses the XML language. Ontologies are used for the standardisation of the messages contents.

The communication mechanisms are grouped in three levels: contents, message and physical levels. The contents level interprets the message and applies the appropriate ontology in order to standardize the data translation. The message level formats the message to send, using an agent communication language, such as FIPA. The physical level allows the physical iteration with other agents and will use CORBA. The advantage to use the CORBA platform is the platform independency (interaction between applications running in distinct platforms, such as Windows, Linux and Unix) and language independency (interaction between applications developed in different programming languages, such as Java and C++) [12].

VI. THE AGENT-BASED CELL CONTROLLER PROTOTYPE BASED ON ADACOR ARCHITECTURE

This section describes the implementation of the ADACOR architecture in the development of an agent-based cell controller prototype for the flexible manufacturing cell of the IDIT platform.

The application system was developed using the Java language, due to its platform independency, in the Windows 2000 operating system, and uses sockets, a TCP/IP mechanism, for the communication between agents, due to its implementation simplicity.

A. *Actors in the System*

In order to simplify the description and representation of the control application it is necessary to define some nomenclature and symbology that represents the several actors in the system. The symbology used to represent the several agent classes is represented in the next figure.

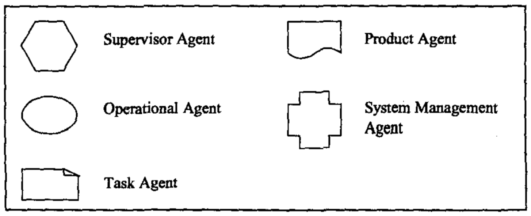


Figure 6 – Agent Classes Symbology

The cell controllers within the flexible platform are represented by Supervisor agents, which will be designated by *MancC* (Manufacturing Cell Controller), *TranscC* (Transportation Cell Controller) and *AsscC* (Assembly Cell Controller). Inside the manufacturing cell, the physical resources are represented by Operational agents, designated by *Turn* (turning center Lealde), *Mach* (machining center Kondia) and *Robot* (robot Kuka).

Each available product in the system is represented by a Product agent, forming a set of four product agents: *Prod_Base*, *Prod_Corpo*, *Prod_Tampa* and *Prod_Pega.*

When it is necessary to produce a product, a Task agent is launched, in order to supervise the execution of the product. In this way, will exist in the system four Task agents: *Base*, *Corpo*, *Tampa* and *Pega*.

B. *Contextualisation of the Control Application*

In this case study, since the focus is the manufacturing cell, will only be considered the products that involve the physical processing of materials.

Additionally, all products require auxiliary operations, such as preparing, fixing and cleaning, which are not considered due to the application specificity.

In this system, each product launches a task agent to supervise its execution. If that final product comprises other sub-product, the task agent will launch other task agents to supervise the execution of each sub-product.

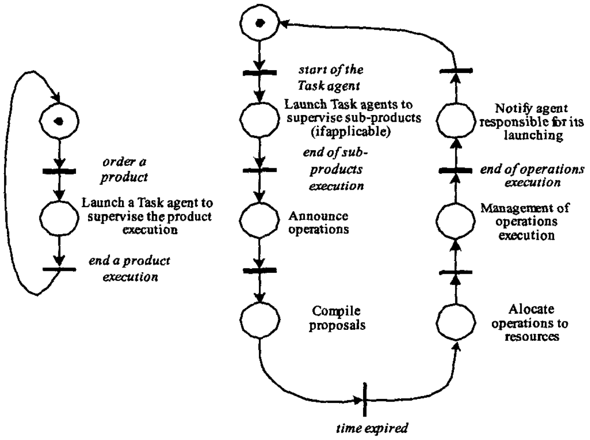

order a product
Launch a Task agent to supervise the product execution
end a product execution
start of the Task agent
Launch Task agents to supervise sub-products (ifapplicable)
end of sub-products execution
Announce operations
Compile proposals
time expired
Alocate operations to resources
Management of operations execution
end of operations execution
Notify agent responsible for its launching

Figure 7 – Model to the Product Life-Cycle

Each task agent should announce all operations to the available operational and supervisor agents, allocate them to the best proposals and wait for the end of the execution of the operation.

C. *Registration and System Management*

In order to interact with other agents in a particular agent system or community, it is necessary that all agents make the registration into the System Management agent that is responsible for the actualised list of available agents in the system.

The existence of an agent that manages the registration in the system can lead to a possible centralisation in the system, because if the System Management agent blinds, all the agents lose the reference to the other agents. In order to avoid this possible centralisation, it is used a decentralised procedure to have the references for the system agents.

Whenever an agent enters in the system, it sends a message to the System Management agent to register in the system and receives a list of actual available agents in the system.

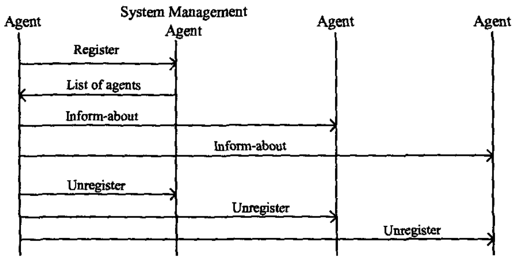


Figure 8 – Registration Schema

The agent stores that list of available agents, becoming independent of the System Management agent. Additionally, it sends a notification message to all active agents in the system indicating its availability. The agents that receive this notification add the new agent to its agents list.

When an agent leaves the system, it sends deregistration messages to all active agents and to the System Management agent. The agents that receive this message should remove the agent from its agents list.

The remaining problem is that if the System Management agent blinds, no agent can register in the system. When the System Management agent gets back to normal functioning, it should consult all agents in its list in order to verify which agents still are active.

D. Task Allocation and Dispatching

The allocation of operations that belongs to a task, the dispatching and their execution, is a crucial aspect in the control application. In this scenario the operations are announced individually, being the task agent responsible for the analysis and allocation of the operations. An important issue to be considered is the precedence between operations, which affects mainly the start date for each operation.

The control of the flexible manufacturing cell entities can be performed by several distinct organisational structures. In this case study, two of them, hierarchical and heterarcical structures, are implemented.

Hierarchical Organisational Structure

In the hierarchical organisational structure, there is a supervisor agent, which takes the name of cell controller, and which is responsible for the coordination of the operational agents that represent the cell resources. Those operational agents are not visible from the exterior, and the task agents can only interact with the supervisor agents.

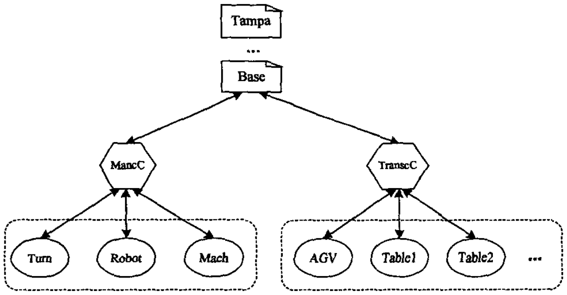


Figure 9 – Hierarchical Organisational Structure

In this scenario, the task agent decomposes the task in operations and announces them to the supervisor agents available in the system. Each supervisor agent (cell controller) verifies the availability to execute the operation and elaborates a proposal to the task agent. The supervisor agent can ask for additional information to the operational agents that coordinate, in case of don't have enough information to elaborate a proposal.

After the expiration time, the task agent takes a decision and allocates the operation to the supervisor agent that had presented the best proposal. The supervisor agent should manage the execution of the operation, through the dispatch of the operation to the operational agent that represents the resource that will execute the operation.

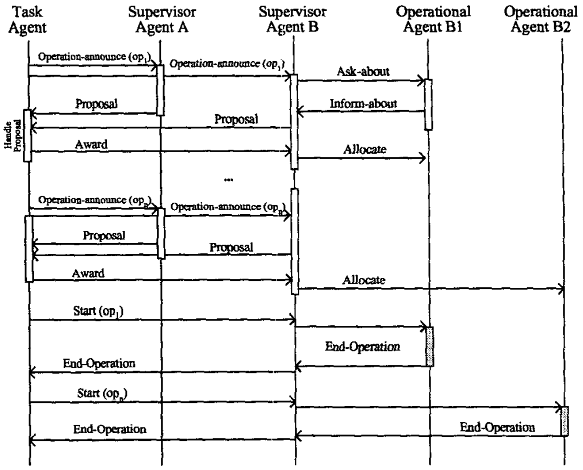


Figure 10 – Communication Between Agents in the Hierarchical Structure

When the operation is finished, the supervisor agent should notify the task agent. The hierarchical organisational approach presents the following features:

- The announcement of the individual operations instead the announcement of the task, doesn't bring the expected optimisation in allocation process.
- The handling with auxiliary operations, such as material transportation from transfer tables to machines, is easier in this approach.
- The pursuing of the plans (provided from other hierarchical supervisor agent) is optimised in this approach.
- Low level of communication in the cooperation process (less announcement messages) but more exchanged messages during the functioning of the system.

Heterarchical Structure

In the heterarchical organisational structure, there isn't a supervisor agent that represents the cell controller, being the cell controller diluted by the several operational agents that represents the cell resources.

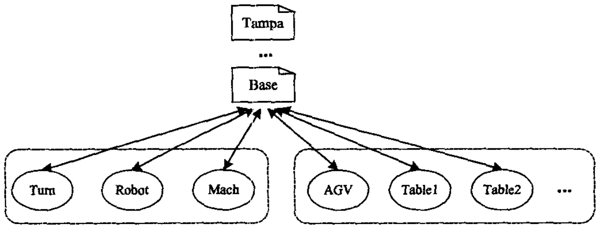


Figure 11 – Heterarchical Organisational Structure

Initially, the task agent announces the first operation to all operational agents available in the system. After the compilation of all proposals, the task agent evaluates and allocates the operation to the best proposal.

The next step is the announcement of the second operation, using the same procedure and indicating a start date based in the end date of the previous operation. This procedure is repeated until all operations are allocated.

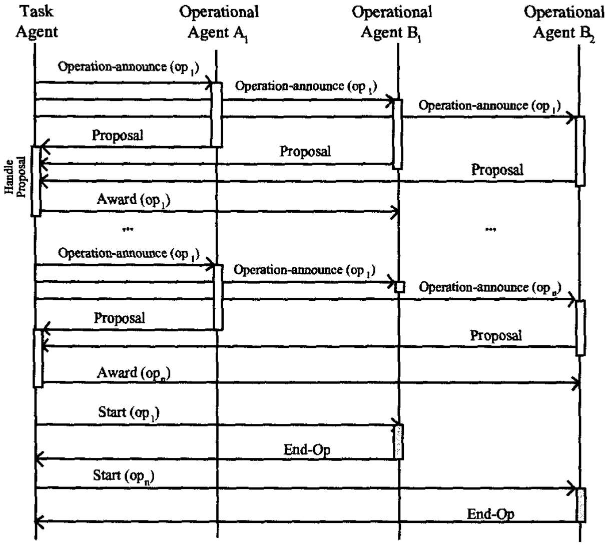


Figure 12 – Communication Between Agents in the Heterarchical Structure

The operational agents can start the execution of the operation after the reception of the *Inform-about* message, indicating the availability to start the operation and the position of the material. The task agent gives this indication after the allocation and execution of the handling operations.

The heterarchical organisational approach presents the following features:

- Reaction to disturbances: easy cooperation with other agents at the same hierarchical level, making easier the resolution of the problems.
- This approach is suitable for environments with high level of autonomy.

Other Announcement Approaches

The announcement approach used in the implementation of hierarchical and heterarchical structures, is a simple and efficient approach, in terms of guaranteeing the temporal sequence and operations precedence. However, it is a sub-optimised approach, because the operations are allocated before the next operations are known. To implement this approach it is used cooperation mechanisms based in the traditional Contract Net Protocol [13].

A more elaborated and complex approach for the announcement process that can be used in both organisational architectures implemented previously is characterized for the simultaneous announcement of all operations. Initially, all the operations are announced simultaneously, indicating the start date, the due date and the precedences that each operation involves. The start date is calculated using the following formula:

$$Op_{i,k} : sd_{i,k} = dd_{i,k} - pt_{i,k} - fg$$

where the *dd* represents the due date, the *pt* the estimated processing time and *fg* the time margin.

Using the *Provisory-Award* message, the operations are allocated temporarily to the resources. However, it is possible that some operations are overlapped, so it is necessary to start a multiple iteration to refine the start and end times for each operation, avoiding the overlapping in the execution of the operations.

When all operations are confirmed, the *Provisory-Award* messages are replaced by the *Award* performatives.

This approach presents the following features:

- More complex implementation approach, involving a larges communication volume and the propagation of the operations precedences,
- Optimisation of the allocation process,
- Cooperation mechanisms that extends the traditional CNP with multiple iteration and temporary allocations.

VII. APPROACH COMPARISON

After the implementation of the agent-based cell controller prototype and with the experience of previous implementation using a traditional approach for the same application, it is possible to compare both approaches and find out the benefits of each approach:

- **Platform independency**: the use of Java language and distributed communications platforms, such as CORBA, to develop control applications, allows the use of the same application in different operating systems environments (such as Windows, Linux and Unix), being platform independent.
- **Application development**: using the agent-based approach, the software necessary to develop the application is shorter and simpler to write, to debug and to maintain.
- **Code re-useability**: the multi-agent technology concept allows an easy and modular development of control applications. Additionally, some components of the developed control application can be re-used for other applications.
- **Distribution and Autonomy**: each agent has autonomy, has control about its behaviour and has local and community knowledge. Is this way, it is possible to build distinct and independent agents that can be placed transparently in a distributed environment.
- **Plugging Intelligence**: the addition of intelligence to a agent, for example to take decisions, manage disturbances

or learning, is a transparent process for the agent and can be viewed as a plug-in of an intelligence module, which takes easier the development of control applications.

- **Specific Applications**: for small or specific applications, the traditional approach can present advantages of small complexity in the code development (in this case it isn't important the re-use of the code and the multiplexing of the developed code).

VIII. CONCLUSIONS AND FUTURE WORK

This paper reflects the research that is being done in the area of distributed and cooperative automation and manufacturing systems, through the use of multi-agent technology to develop agile control systems that fulfil the main requirements presented by that kind of applications, such as distribution of applications, cooperation, self-organisation, and integration of humans and physical devices.

In this paper it was presented an agent-based architecture approach to the development of distributed manufacturing applications and its implementation into a agent-based manufacturing cell controller. In comparison with the traditional approaches, the agent-based approach presents important improvements such as expansibility, robustness, reactivity, support to distributed environments and re-use of the application code.

The prototype developed for the case study, presented in the paper, was developed in Java, using a proprietary communication protocol and a non-optimised existing application. The next phase of research is to improve cooperation, decision and adaptation mechanisms and the agent-based platform, through the use of an agent development tool, such as JADE (Java Agent Development Framework), which is FIPA compliant and has an easy connection to Jess (*Java Expert Shell System*) [12]. Another important improvements are the introduction of more intelligence in each agent, the development of interfaces to physical devices and the application to more complex case studies.

Additionally, it is important to develop a metrics methodology to allow the comparison of the traditional and agent-based control approaches.

IX. REFERENCES

[1] R. Rabelo and L. Camarinha-Matos, "Multi-Agent based Dynamic Scheduling", *Int. Journal on Robotics and Computer Integrated Manufacturing, Vol II*, n.4, 1994, pp 303-310.

[2] F. Maturana and D. Norrie, "Multi-Agent Mediator Architecture for Distributed Manufacturing", *Journal of Intelligent Manufacturing, Vol. 7*, 1996, pp 257-270.

[3] H. Van Dyke Parunak, A. Baker and S. Clark, "The AARIA agent architecture: from manufacturing requirements to agent-based system design", *in Workshop on Agent-based Manufacturing, 1998,* Minneapolis.

[4] W. Shen, D. Xue and D. Norrie, "An Agent-Based Manufacturing Enterprise Infrastructure for Distributed Integrated Intelligent Manufacturing Systems", *in Proceedings of the 3rd Int. Conference on the Practical Application of Intelligent Agents and Multi-Agents*, London, UK, March, 1997, pp 23-25.

[5] T. Heikkilä, M. Kollingbaum, P. Valckenaers and G.-J. Bluemink, "manAge: An agent architecture for manufacturing control", *in Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, K. U. Leuven, 1999, pp 127-136.

[6] A. Quintas and P. Leitão, "A Manufacturing Cell Controller Architecture", *in Proceedings of Flexible Automation and Intelligent Manufacturing Conference*, Middlesbrough, 1997, pp 483-493.

[7] ISO/IEC 9506, *Industrial Automation Systems - Manufacturing Message Specification*, Part 1 - Service Definition, 1992.

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

[9] W. Shen and D. Norrie, "Agent-based systems for intelligent manufacturing: A state-of-the-art Survey", *Knowledge and information systems: an International Journal, 1* (2), 1999, pp 129-156.

[10] H. Van Dyke Parunak, *What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC*; Industrial Technology Institute; 1998.

[11] P. Leitão and F. Restivo, "An Agile and Cooperative Architecture for Distributed Manufacturing Systems", *in Proceedings of the IASTED International Conference Robotics and Manufacturing,* 21-24 May, 2001, Cancun, Mexico, pp 188-193.

[12] J. Barata, L. Camarinha-Matos, R. Boissier, P. Leitão, F. Restivo and M. Raddadi, "Integrated and Distributed Manufacturing, a Multi-agent Perspective", *in Proc. of 3rd Workshop on European Scientific and Industrial Collaboration*, 27-29 June, 2001, Netherlands.

[13] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Solver", *IEEE Transactions on Computers*, Vol. C-29, N°12, 1980, pp 1104-1113.