ACTA TECHNICA CORVINIENSIS – Bulletin of Engineering
Tome IV (Year 2011). ISSN 2067-3809

[1.] Renato VIDONI

A MULTI AGENT ROBOTIC SYSTEM FOR SIMULATION AND CONTROL OF A MANUFACTURING PROCESS

ABSTRACT:
In this work the multi-agent technology is exploited in order to develop a Multi-Agent Robotic System with the aim to simulate and control a production chain and lays the bases for the introduction of the agent technology into a manufacturing industrial process.
In particular, a simplified washing-machine production system has been studied and agentified. More in detail, automatic production, negotiation, supplying of pieces and management of the production have been considered.
The overall simplified system has been implemented by means of the JADE (Java Agent Development Environment) platform, compliant with the FIPA (Foundation for Intelligent Physical Agents) specifications, and extensively tested in order to prove the robustness and effectiveness of the approach. The developed simplified system has been conceived in order to be easily expansible, thanks to its modularity and structure, and ready to be upgraded.

KEYWORDS:
manufacturing process, assembly line, Multi-Agents, robots

INTRODUCTION

Reconfigurable and adaptive production systems, which can provide companies with the proper level of agility and effectiveness, are necessary in order to satisfy fast changes of customers' needs and demands.
Markets are highly competitive and push manufacturing systems from a mass production to a mass customization fashion. A reduction of the product life-cycles, short lead times and high utilization of resources without increasing the costs are the main targets to satisfy.
In order to comply with these requests, Just In Time (JIT) techniques that allow to reduce waste of time and resources are adopted. Thus, the market is becoming more and more mutable and changeable and new technologies have to be adopted in order to react and adapt in a fast manner.
Such premises do not allow a centralized production because a high amount of work and a low flexibility of the structure will occur. Hence, production means need to become reconfigurable and founded on autonomous and intelligent modules, which dynamically interact with each other for the achievement of local and global objectives. Production processes have to provide the required level of agility, i.e. the ability to success in a rapidly changing outer environment, and embed adaptivity attributes. Moreover, in a flexible production system the goals are also the time-to-market reduction, the raise of the productivity level and the cost reduction.
Autonomous and intelligent agents - an agent is a system situated in some environment and capable of autonomous action in this environment in order to meet its design objectives [1] -, that model the production by means of a decentralized control unit and are suitable for high uncertainty and error ratios, can be the answer. Indeed, differently from the Computer Integrated Manufacturing, there is no need of a centralized approach and a unique complex process manager; only the communication and negotiation phases between agents for the use and control of machinery, resources and materials are needed.
The main advantages of this technology and approach are:
1. Decentralized and distributed decision (i.e. each agent keeps decisions autonomously).
2. Modular structure (i.e. agents are independent).
Agents can be used and exploited for:
1. Simulation. Agent frameworks are extensively used where the interactions between different entities have to be studied;

2. Management and control. An effective control system has to show flexibility, fault tolerance, reusability and low costs. As underlined in [2], agent technology is suitable for effectively reacting to the production changing and high volumes production.

In the last decade, the scientific community *has* contributed *to the development of techniques and applications of Multi-Agent systems (MAS [1,3]).*

A Multi-Agent System (MAS) consists of a group of agents that can potentially interact with each other [3]. By exploiting this feature several advantages such as reliability and robustness, modularity and scalability, decentralization, time-dependency, adaptivity, concurrency, parallelism and dynamism [4,5] can be reached. Bussman in [6] shows how a Multi-Agent system matches the requirements for agile and fast reaction to sudden and unpredictable changes in production demands and is suitable for high volumes production.

In literature, many works that deal with MAS can be found (e.g. [2]; [7]; [8]; [9]; [10]) but agent concepts and techniques are rarely applied and practically adopted in industry. Only few applications and a small part of the available technology has been successfully applied and is currently on the market. How underlined in ([2]; [11]), many works do not specify the working environment or the production plant, focusing only on the definition of a general or theoretical model. Usually, MAS has been applied in order to simulate process flows or decision activities by developing demonstrators, industrial process and chain production simulators, and small system prototypes ([8]; [12]; [13]; [14]; [15]).

At today, industrial companies rarely use Multi-Agents mechatronic systems in production and management. Most of the works are only simulations ([2]; [11]; [16]) while successful working applications is low. Among these, the flexible and distributed MAS control of a ship equipment, the Rockwell Automation, Inc. "MAST" simulation tool for material-handling and the application of MAS for production planning of SkodaAuto cars [11] can be cited.

With the purpose of standardizing agent technologies for the interoperation of heterogeneous software agents, "the Foundation for Intelligent Physical Agents" (FIPA) has become an IEEE Computer Society standards organization and has developed specifications for permitting the creation of a set of shared rules. Thanks to this efforts on standardization, FIPA-OS (FIPA-Open Source), JADE (Java Agent Development Environment) and ZEUS agent platforms compliant to the FIPA rules and directives have been created. The JADE agent platform [17], based on Java, has been chosen in this attempt in order to implement the agents and deploy the multi-agent environment.

In this work, the intelligent agent techniques will be adopted in order to study and develop a distributed framework in order to simulate and control an industrial process like the washing machine chain production.

The target is to model and identify the production chain and to realize a Multi-Agent-Robotic-System (MARS) able to simulate and control a fully autonomous assembly process that works by means of the cooperation of software and robotic agents. Such physical robotic agents move on the environment in order to comply with specific requests as transport and station restocking like in a real factory.

THE STUDY: WASHING-MACHINES CHAIN PRODUCTION

Industrial manufacturing, robotized and autonomous operations and washing-machines chain production are addressed in this work. These fields suit well with the agent technology and theories.

The main target is to develop a MARS (Multi Agent Robotic System) with a high degree of flexibility that can be exploited as the base for a future intelligent and automatic system able to control the full chain production. Such a chain can be viewed, in a simplified scheme, as a collection of working stations.

Each station is in charge to assembly one (or a set of) component and pass the piece at the following station where the next correct component has to be assembled.

Each station is furnished with a small store of components (i.e. local store) that, in order to not stop the production, has to be correctly supplied and refurnished. Thus, each station can be viewed as an agent that is autonomous and reactive. Moreover, autonomous agents or robotic agents can be applied in order to consider and substitute the human work (i.e. a human that drives a forklift truck for supplying an order).

In practice, when the autonomous agent that represents the i-th station realizes that the local store has to be refurnished, searches a free autonomous robotic agent that can supply its request both in terms of material quantity and time. After a negotiation phase and the entrust of the task, the chosen autonomous robotic agent goes to the central store, loads the components and brings them to the calling station. Some agents can be used as traders and facilitators in order to manage the communication or settle the conflicts and some others can record the robot data in order to give information to the stations like a yellow page service.

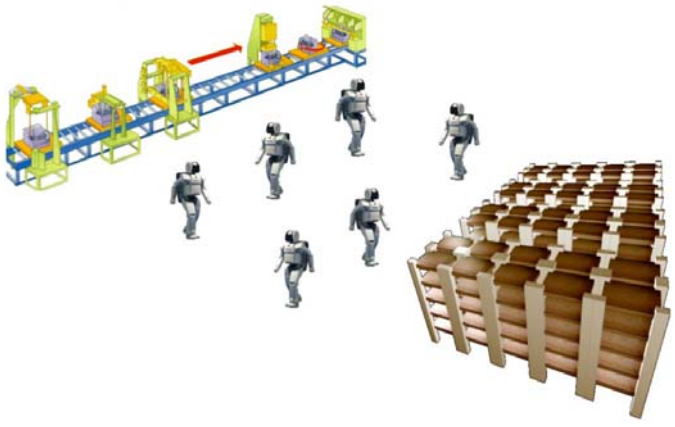


Fig. 1: Sketch of the scenario: assembly line, supply robots and main store

In this attempt, some simplifications have been made but the developed framework is implemented in order to be modular and easily expanded. A sketch of the chosen scenario is shown in Fig. 1.

In this work the simplified supply chain has been modeled by means of four stations, each one in charge of assembly a specific part of the washing-machine:

- *Station 1: motor and belt;*
- *Station 2: drum and washing tank ;*
- *Station 3: two bearings;*
- *Station 4: frame.*

In a simplified view, the production follows the following operations:

- *Station i, if the component is available in the local store, assembles the piece and, if Station i+1 is free, pushes forward the assembly; if Station i does not have the piece to be mounted, searches and calls an autonomous robotic agent in order to be re-furnished. If an autonomous robotic agent is free, it is charged/entrusted to go the the central store, keep the lacking components and refurnish the calling station.*
- *When Station i has sent the current assembly to the station i+1, its internal state is set on "free" and it is able to receive and manage another piece.*

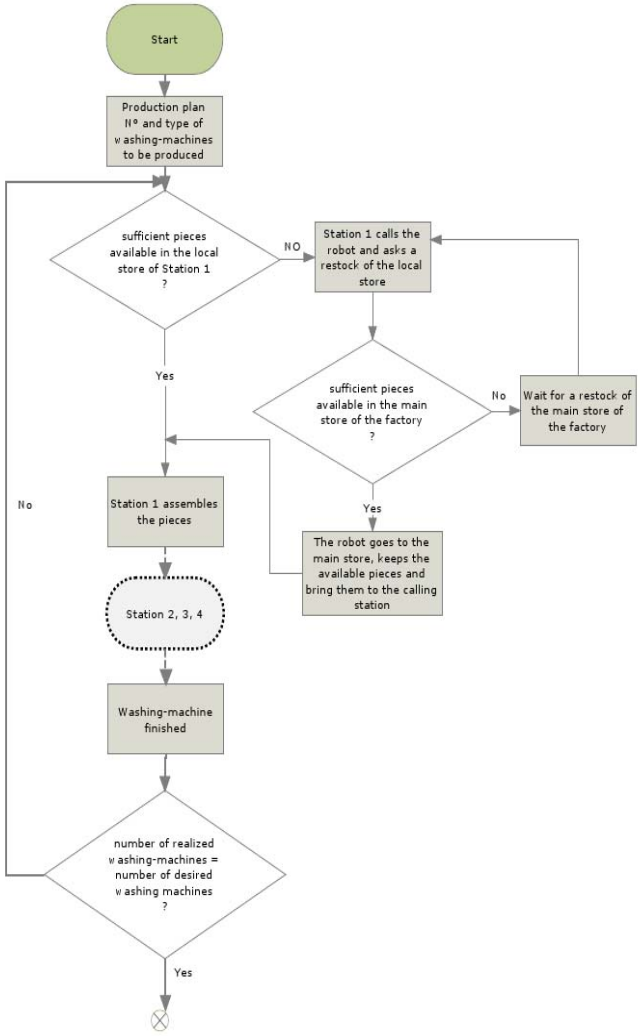


Fig. 2 : Flow-chart of the simplified process

All the works and operations have to be carried out in a parallel and not sequential manner. The flow-chart of the simplified process is presented in Fig. 2.

THE MA(R)S SYSTEM

At least two kinds of agents are necessary: one for the stations, Station Agent, and one for the autonomous robots, Robot Agent.

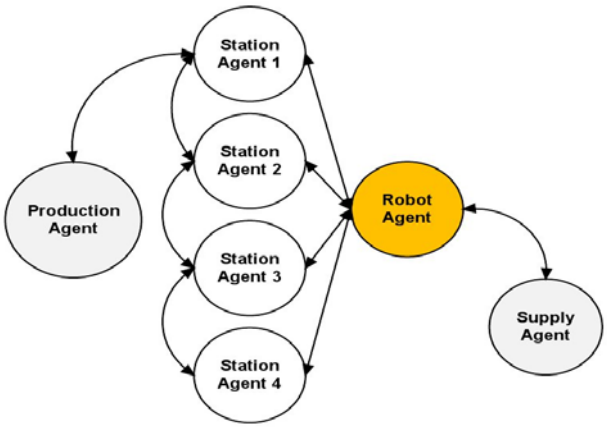


Fig. 3 : Agents of the framework

Moreover, a Production Agent, that is in charge to start and stop the process flow, and a Supply agent, that will manage and supply the main store when the stocks are ending, are necessary. Thus, by considering for sake of simplicity only one autonomous robotic agent, 7 agents are single out: 4 Station Agents, 1 Production Agent, 1 Robot Agent and 1 Supply Agent.

If all works properly, the Production Agent calls Station Agent 1 that, after its assembly task, calls Station Agent 2 and so on. If a component to be mounted is ending, the related station calls the Robot Agent that negotiates the components with the Supply Agent (see Fig. 3).

Production Agent

The Production Agent starts the production and sets the number of washing-machines that has to be produced.

First of all, it continuously searches all the available Station Agents by means of a cyclic behavior.

Inner this behavior, the Station Agent 1, i.e. the one related to the first station, is searched by means of a CALL_FOR_PROPOSAL (CFP)-message inner the StationCommand behavior.

If the Production Agent receives a PROPOSE-message, the Station Agent 1 is ready to work; otherwise, if the replies are only REFUSE-messages, it means that there are no free agents or their name is not Station Agent 1 and the behavior is repeated (general case). Within the REFUSE-MESSAGE the number of completed washing-machines is passed and, in the particular case of a number of completed pieces equal to the desired one, the production is completed.

If a PROPOSE-message is received, the Production Agent sends an ACCEPT_PROPOSAL-message with the current number of lacking washing-machines to the Agent Station that replied with a PROPOSE-message.

Station Agent

Each Station Agent has to know the number of available pieces in its local store, the part to assembly, how many and which Robot Agents and Station Agents are present in the framework and the name of the following station.

Each station has a certain number of available pieces in its local store and of elements that has to assembly (e.g. motor, bearings).

Moreover, each agent has to register itself into the Directory Facilitator (DF) that is a sort of "yellow page", in order to be found.

For this class, six behaviors are implemented:

· ResearchRequest, cyclic.

This behavior waits the CFP-messages from both the Production and the other Station Agents and sends a PROPOSE-message if it is free and is the correct station. Otherwise, a REFUSE-message is sent.

· ArrangeResearch, cyclic

This behavior starts with the ACCEPT_PROPOSAL-message of the Production Agent and sends an INFORM-message to the calling agent.

· ProductionResearchRobotAgent, one-shot.

It is called inner the ArrangeResearch behavior. If the local store is not empty, the correct piece is assembled and the SearchStationAgent behavior is called. If the local store has few pieces or it is empty the ExecutionResearch behavior is called.

· ExecutionResearch, generic.

It asks the lacking components to a suitable Robot Agent.

· SearchStationAgent, oneshot.

It is similar to the cyclic behavior of the Production Agent. It calls the StationbystationCommand generic behavior.

· StationbystationCommand, generic.

Similar to the CommandStation behavior. The next station is searched in order to pass the current washing-machine to be assembled.

If the local store is empty, a REQUEST-message is sent to the Robot Agents. A first research of the available robots is made; after that the lacking piece is requested by means of a CFP-message.

The negotiation is based on the time requested to supply the local store. The request is repeated until when at least one robot replies positively.

Supply Agent

It has two cyclic behaviors:

· SupplyRequest, cyclic.

In this behavior the supply requests from the Robot Agents are evaluated.

In this first version of the framework the main store is considered as ideal and, thus, the reply is always a PROPOSE-message.

· SupplyOrder, cyclic.

With this behavior the supply requests are satisfied and an INFORM-message is sent to the calling Robot Agent when the operation is ended.

Robot Agent

Each Robot Agent is able to supply a certain number of pieces and is registered into the DF in order to make visible its services. Moreover it represents a real autonomous robotic system (e.g. AGV, forklift truck).

The implemented behaviors are:

· RequestOffer, cyclic.

This behavior is used in order to supply the CFP-messages of the Station Agents. If the requested components are available and the robot is able to fulfill the order, it replies with a PROPOSE-message with the estimated time for making the operation. Otherwise a REFUSE-message is sent.

· SupplyOrder, cyclic.

This behavior reacts when the Agent Station accepts the proposal; it makes the restocking and sends an INFORM-message when a successful delivery is done.

· SearchSupplyAgent, one-shot.

It searches a Supply Agent in order to satisfy the order. It calls the SupplyOrder behavior.

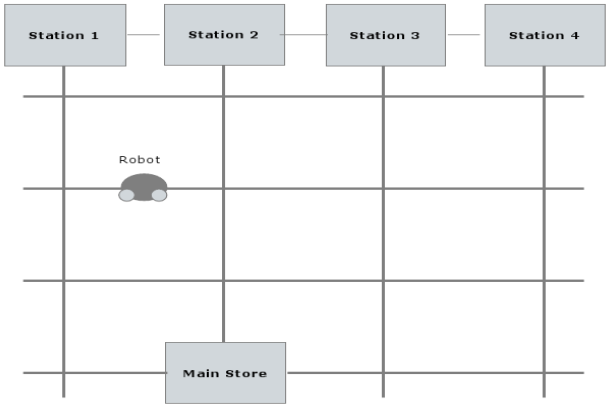
· SupplyOrder, generic.

Similar to StationCommand behavior.

When a Robot Agent accepts a task, its state is set to "occupied" until when the requested components are not delivered.

In order to realize a framework that deals not only with software agents but also with physical agents, real robots have to be integrated into the framework. The software language, the communication channel and the physical sensors and actuators have to be set-up.

(a)



(b)

Fig. 4: the NXT robot (a) and the realized scenario (b)

The hardware chosen in order to create a realistic simplified scenario is the Lego Mindstorms NXT shown in Fig. 4(a) [18].
This robot has been employed on a three wheeled (tricycle) configuration with two motors that control the two tractor wheels (see Fig.4(a)). The robot is equipped with different sensors. In particular, two classes of them have been used. Two light sensors have been installed and located at the robot front in order to follow the chosen road and recognize the crosses. Indeed the travel area has been defined as a grid of roads to be followed. Also, an ultrasonic distance sensor has been included in the robotic system in order to evaluate if there are obstacles in a range between 10 and 15 cm. Java is the programming language that has been used to implement the overall system. In particular, in order to command and control the robot, the ICOMMAND API has been exploited and the leJOS firmware has been employed [18].
As depicted in Fig.4(b), the working stations are at the top while the main store is at the bottom of the plant. When a robot is called and accepts the work, it has to leave from the calling station or from the current position, go the central store and come back to the calling station. For each station a predefined path can be used or a shortest path searching algorithm can be exploited. In order to communicate, send orders and receive information with the real robot, a Bluetooth-connection is established between the Robot Agent and the NXT.
The overall system has been implemented on an Intel Pentium Dual Core T3400 @ 2.16 Ghz, RAM 4 GB PC hardware and extensively tested in order to fix bugs or unwanted behaviors. Different production orders have been sent to the Production Agent in order to simulate a realistic production.

CONCLUSIONS

In this work a Multi-Agent Robotic System has been studied, defined and realized in order to lay the bases for the optimization and management of an industrial process by means of the theories based on the autonomous agents.
As a scenario, a washing-machine assembly line has been evaluated. Human operators that work in order to restock the local stores of each working station have been considered as autonomous (robotic) agents.
Each working station has been modeled as an agent, and a simplified agentification of the process has been made. The overall framework and agents have been realized by means of the JADE platform that follows the FIPA standards and a NXT robot has been integrated into the system in order to simulate in a realistic manner the motion of a robotic agent and the communication process into the real-environment.
Future work will cover the extension of the framework in order to consider the main store management and the integration of different robotic systems with high navigation, motion and handling capabilities.

REFERENCES

[1.] *Wooldridge, M. An introduction to multiagent Systems. Ed. JohnWiley & SonsLtd., Chichester, (2002).*
[2.] *Caridi, M. and Cavalieri, S., (2004), Multi-agent systems in production planning and control: an overview, Production Planning & Control, 15:2, 106-118.*
[3.] *Vlassis, N. A concise introduction to multiagent systems and distributed artificial intelligence. Synthesis, Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers. (2007).*
[4.] *Elamy, A. H. Perspectives in agent-based technology. AgentLink News, Vol. 18 pp. 19-22. (2005).*
[5.] *Parunak, H. V. D., (1998), Practical and industrial applications of agent-based systems. URL: http://agents.umbc.edu/papers/apps98.pdf.*
[6.] *S. Bussmann, K. Schild, An agent-based approach to the control of flexible production systems, In Proc. of the 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETFA 2001) (2001), pp. 481-488.*
[7.] *Hao, Q, Shen, W, Zhang, Z, (2005), An Autonomous Agent Development Environment for Engineering Applications, Int. J. of Advanced Engineering Informatics, 19(2):123-134.*
[8.] *Hao, Q. and Shen, W., (2006), An agent-based simulation of a JIT material handling system, in IFIP Int. Federation for Information Processing, 220: 67-78, Information Technology for Balanced Manufactiuing Systems, (Boston: Springer).*
[9.] *Sayda, A.F., and Taylor, J.H., (2007), Toward A Practical Multi-agent System for Integrated Control and Asset Management of Petroleum Production Facilities,IEEE Int. Symposium on Intelligent Control (ISIC), Singapore, October 2007.*
[10.] *Dangelmaier, W, Heidenreich, J, Pape, U, () Supply Chain Management: A Multi-Agent System for Collaborative Production Planning, Proc. Of the IEEE Int. Conf. on e-Technology, e-Commerce and e-Service, 2005. (EEE '05).*
[11.] *Pechoucek, M., Marík, V, (2008), Industrial deployment of multi-agent technologies: review and selected case studies, Autonomous Agent and Multi-Agent Systems, 17:397–431.*
[12.] *Albert, M., Längle1, T., Wörn, H., Capobianco, M., Brighenti, A., (2003), Multi-Agent Systems for Industrial Diagnostics, Proceedings of 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes 2003.*
[13.] *Hallenborg, K, (2007), Decentralized scheduling of baggage handling using multi-agent technologies, in Multiprocessor Scheduling: Theory and Applications, ed. Levner, Dec. 2007, Itech Education and Publishing, Vienna, Austria*

[14.] Trullàs-Ledesma, J. and Ribas-Xirgo, L., (2008), Integrating physical and software sub-systems in a manufacturing environment through agentification, IX Workshop of Physical Agents 2008.
[15.] Xiaohua Liu, Jie Lin, Feng Wang, (2009), Simulation system of Production Scheduling Multi-Agent-Based, 2009 World Congress on Computer Science and Inf. Eng.
[16.] Burmeister, B, Industrial Application of Agent Systems: Lessons Learned and Future Challenges, L. Braubach et al. (Eds.): MATES 2009, LNAI 5774, pp. 1–3, 2009. © Springer-Verlag Berlin Heidelberg 2009.
[17.] Bellifemine, F., Caire, G., Poggi, A., Rimassa, G. JADE: A software framework for developing multiagent applications. Lessons learned. Information & Software Technology, Vol. 50(1-2) pp. 10-21, 2008.
[18.] leJOS, 2010. Java for lego mindstorms. Web available, http://lejos.sourceforge.net

AUTHORS & AFFILIATION

1. Renato VIDONI

1. DIPARTIMENTO DI INGEGNERIA ELETTRICA, GESTIONALE E MECCANICA UNIVERSITA' DI UDINE, UDINE – ITALY

ACTA TECHNICA CORVINIENSIS
– BULLETIN of ENGINEERING
ISSN: 2067-3809 [CD-Rom, online]