Multi-agent system for dynamic scheduling

1st Bernardo Firme
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
bernardo.firme@tecnico.ulisboa.pt

2nd Guilherme Lopes
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
guilherme.cordeiro.lopes@tecnico.ulisboa.pt

3rd Miguel S. E. Martins
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
miguelsemartins@tecnico.ulisboa.pt

4th Tiago Coito
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
tiagoascoito@tecnico.ulisboa.pt

5th Joaquim Viegas
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
joaquim.viegas@tecnico.ulisboa.pt

6th João M. C. Sousa
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
jmsousa@tecnico.ulisboa.pt

7th João C. P. Reis
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal
joao.c.p.reis@tecnico.ulisboa.pt

8th João Figueiredo
Department of Physics
Universidade de Évora
Évora, Portugal
jfig@uevora.pt

9th Susana Vieira
IDMEC, Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
susana.vieira@tecnico.ulisboa.pt

***Abstract*—This paper proposes a flexible manufacturing system based on intelligent computational agents. A Multi-Agent System composed of 4 types of reactive agents was designed to control the operation of a real implementation in the Intelligent Automation Lab at Instituto Superior Técnico. This implementation was based and constructed analogously to a known benchmark, AIP-PRIMECA. The agents were modelled using Petri nets and agent communications were defined through the combination of FIPA Interaction Protocols. The system was tested under the conditions of static and dynamic scenarios, having its performance validated whenever possible by comparison with results from a Potential Fields Approach in the same benchmark. Overall, the performance exhibited by the proposed MAS was slightly better and it is worth highlighting the simple behaviour of each agent and ability to respond in real-time to all the dynamic scenarios tested.**

***Index Terms*—Multi-Agent Systems, Agent Communication, Manufacturing Control, Flexible Manufacturing System**

I. INTRODUCTION

The industry has been evolving to provide products with increasingly high-quality and customization, having already faced three revolutionary stages known as industrial revolutions. With the introduction into the manufacturing world of emerging technologies like Internet of Things, wireless sensor networks, Big Data, cloud computing and Artificial Intelligence, a new paradigm arrived to the industry world: the fourth industrial revolution [1].

Aiming the achievement of a smart, flexible and reconfigurable factory, capable of producing customized and small-lot products efficiently and profitably, the use of traditional manufacturing control systems is not enough. These control systems do not exhibit sufficient capabilities of responsiveness, flexibility and reconfigurability, since they are designed based on centralized and hierarchical control structures that, despite presenting good production optimization, present insufficient response to change due to rigidity and centralization. To respond to this handicaps and taking advantage of the emergent technologies, some advanced manufacturing control systems have been proposed to achieve the smart factory of the future. [2]

A representative case of the advanced manufacturing control systems is the agent-based manufacturing control based on Multi-Agent Systems (MAS) technology. MAS consist in an ecosystem of manufacturing resources defined as intelligent, autonomous and cooperative computational entities, known as agents, that can negotiate with each other to implement decision making and dynamical reconfiguration, in order to achieve their individual goals which, in aggregate, accomplish an overall objective. In an agent-based manufacturing control system, all the agents are in the same hierarchy level, being organized in a autonomous, distributed and decentralized architecture [2].

Throughout the last decades, several different approaches, architectures and platforms regarding MAS have been introduced and a considerable amount of industrial applications were already implemented. The main fields of application have been smart production, smart electric grids, smart logistics and smart healthcare [3], although some authors also have a prospect of other fields that might benefit from the application of agent technologies, namely traffic control, buildings and home automation, military and network security [4]. Some recent application of MAS can be found in [16] for optimal energy management and in [17] for management of traffic

---

This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020

978-1-7281-6926-2/20/$31.00

flows in a network.

Regarding smart manufacturing solutions, different authors already extensively surveyed the major approaches and implementations presented so far, citing chronologically [5], [4], [2], [6], [7], [8]. Despite several approaches and implementations, a long path still needs to be travelled in order to have distributed agent-based approaches fully adopted by industry. The approaches with a full industrial implementation are scarce and the implemented functionalities are limited.

The main focus of this work is the design and the real-implementation of a MAS in Intelligent Automation Lab (IAL) at Instituto Superior Técnico (IST) in order to study the applicability of an agent-based manufacturing control.

II. MANUFACTURING SYSTEM

The flexible and reconfigurable manufacturing system to be controlled must be properly defined in order to implement an agent-based manufacturing control system. Thus, there are some required characteristics in its physical configuration:

1) The conveyor system needs to provide more than one path to travel between the same two points, so that the system can provide material-handling flexibility and machine sequence flexibility.
2) Redundancy is necessary to provide flexibility in the products' machine sequence and reconfiguration of the same in case of machine breakdown.

A. AIP-PRIMECA Flexible Manufacturing System

The AIP-PRIMECA Flexible Manufacturing System (FMS), located in the AIP-PRIMECA Center at the University of Valenciennes, was defined as the benchmark [9] since fulfils the necessary requirements and is widely used for research purposes.

This flexible production cell is depicted in Figure 1. Its conveyor system configuration composed of a main loop, four transversal sections composing multiple inner loops responsible for the material handling flexibility and several derivations to reach the machines and positioning units in front of machines allows a really flexible routing of jobs inside the production cell, and the existence of three robots, which provide some operations in common, creates the necessary redundancy for the production.

In terms of production, the smallest elements present in the cell are five components: (1) ”Axis”, (2) ”I_comp”, (3) ”L_comp”, (4) ”r_comp” and (5) ”S_comp”, which combined in a ”Plate” where they are placed, it is possible to assemble 7 different letters: ”B”, ”E”, ”L”, ”T”, ”A”, ”I” and ”P”. The final products proposed to the client are words formed with these jobs and they are three: ”BELT”, ”AIP” and ”LATE”. In this assembly cell, there are eight manufacturing operations and the cell is composed of seven machines, two of which being optional and not used in this work. The machines are represented in Figure 1 with the symbols M1 to M7, being:

- M1: loading and unloading unit
- M2, M3 and M4: assembly workstations
- M5: automatic inspection unit

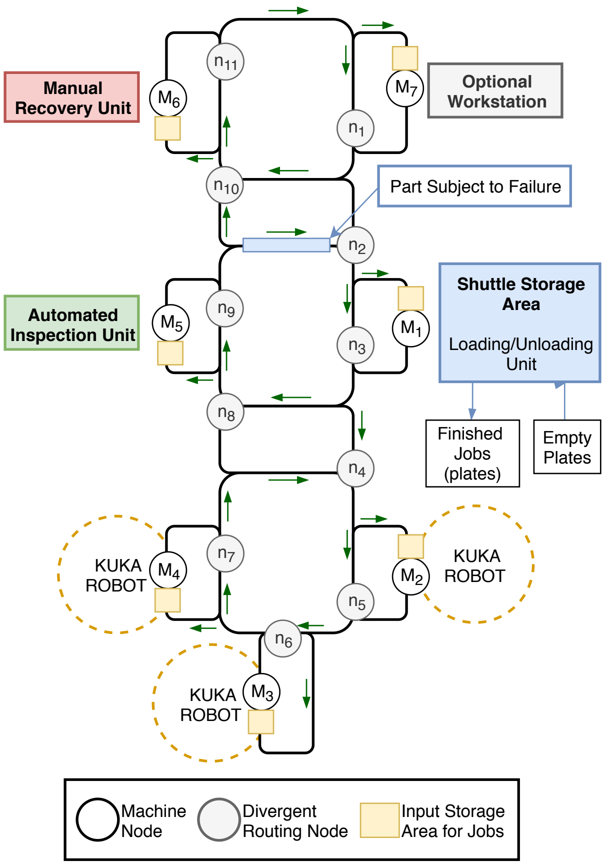

Fig. 1. AIP-PRIMECA cell layout (based in [9])

- M6: manual recovery unit (not used)
- M7: extra assembly workstation (not used)

Table I shows the different operations executed by each machine (M1-M5), together with the corresponding manufacturing processing time of each operation and production sequence for each type of job.

B. Proposed Manufacturing System

The benchmark chosen already has the required flexibility and complexity for the design of an agent-based manufacturing control system and the IAL was designed analogously to the AIP-PRIMECA FMS, so that direct comparison of performance can be carried out.

Among the components available in IAL (which can be consulted in fully detail at [10]), for this work was used a programmable logic controller (PLC) Siemens S7-1500 connected by PROFINET to a SCADA-PC responsible for the supervision of the manufacturing system. Also, an independent PC-Platform, containing the agents' platform, was connected to the PLC through Open Platform Communications - Universal Architecture (OPC-UA). For PLC and SCADA was used TIA Portal V15.1 and TIA Portal V15.1 Runtime, respectively. The agents' platform chosen was Java Agent DEvelopment Framework (JADE) which is one of the most widely used platforms for agent-related research purposes. All components

TABLE I
MANUFACTURING OPERATIONS PROCESSING TIMES AND JOBS PRODUCTION SEQUENCE

No.	**Operations**	**M1**	**M2**	**M3**	**M4**	**M5**
1	Axis	-	20	20	-	-
2	r_comp	-	20	20	-	-
3	I_comp	-	-	-	20	-
4	L_comp	-	20	-	20	-
5	S_comp	-	-	20	20	-
6	Inspection	-	-	-	-	5
7	Plate loading	10	-	-	-	-
8	Plate unloading	10	-	-	-	-

Job	**Production sequence for each type of job**
B	(7)-(1)-(1)-(1)-(2)-(2)-(3)-(5)-(6)-(8)
E	(7)-(1)-(1)-(1)-(2)-(2)-(4)-(6)-(8)
L	(7)-(1)-(1)-(1)-(3)-(3)-(5)-(5)-(6)-(8)
T	(7)-(1)-(1)-(2)-(4)-(6)-(8)
A	(7)-(1)-(1)-(1)-(2)-(4)-(5)-(6)-(8)
I	(7)-(1)-(1)-(3)-(5)-(6)-(8)
P	(7)-(1)-(1)-(2)-(4)-(6)-(8)

and software used together with the communications protocols are represented in figure 2.

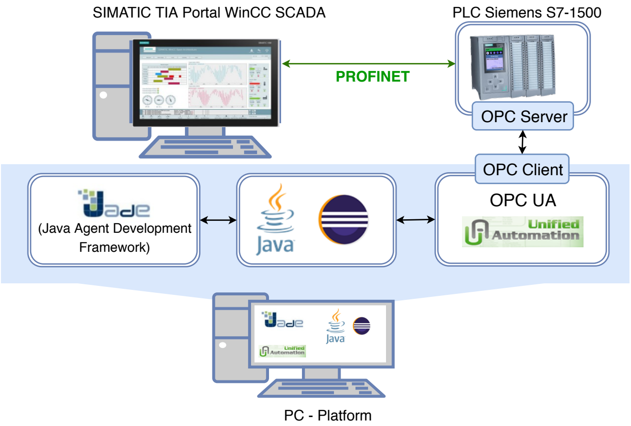

Fig. 2. Components, software and communications from AL 4.0

Nonetheless, IAL doesn't contain enough physical devices to fully reproduce the AIP-PRIMECA FMS so the same were replaced by function blocks created in PLC in order to reproduce the time each physical device's action takes. To carry on the experience, a group of PLC variables designated "Flags" was created to define the communication channels between the agents and the PLC.

Each component reproduced, workstations and shuttles that travel in the conveyor, is represented as a timer and two flags. Flag "ComponentX" is a binary flag that activate the timer and flag "ComponentX_time" represents the time the operation takes. For workstations the correspondent flag for time represent the duration of determined operation (table I) while for shuttles represent the travel time in the conveyor between nodes and workstations (figure 1 and in detail at [9]).

C. Agents

The agent-based model was built according to the physical mapping method, by which different agents are used to represent different physical entities. Taking this into consideration, the designed MAS is composed of four different types of reactive agents:

1) **Order Agents** - Orders submitted by the clients
2) **Job Agents** - The jobs that are loaded into the production system, necessary to complete the orders
3) **Workstation Agents** - Seven workstations representing machine and waiting area for workstations 1,5,6,7 and machines, waiting area and robot in the case of worskations 2,3,4 (figure 1)
4) **Conveyor Agent** - Conveyor belt, responsible to coordinate the shuttles

D. Agent Communication

The design of the MAS was performed according to the Foundation for Intelligent Physical Agents (FIPA) Specifications, in order to take advantage of the standardized communication protocols, message transport and agent management [13]. Agent Unified Modelling Language (AUML) was used to specify and represent these interactions between agents in terms of interaction protocols [14]. The communication protocols between different type of agents are summarized in table II.

TABLE II
AGENTS FIPA COMMUNICATION PROTOCOLS

Agents	**FIPA Communication Protocols**
Order Agent - Order Agent	-Propose Protocol
Job Agent - Order Agent	-Inform (message only)
Job Agent - Conveyor Agent	-Request Protocol -Inform (message only)
Job Agent - Workstation Agent	-Contract Net Protocol -Request Protocol

E. Agents Behaviour

The behaviour of each type of agent was modelled by using the Petri net formalism, which is a tool fit to model and to analyse the behaviour of complex event-driven systems.

An Order Agent (figure 3) is created for each existing order in the beginning of the production and is responsible for the creation of Job Agents that compose it.

Each Order Agent executes a propose protocol with the other Order Agents, so that, through a simple negotiation algorithm, Orders can organize themselves and internally define the order in which they will be produced. In this work, Orders negotiate through the order they are submitted (see further in table III). When the negotiations finish, the Order Agents sequentially create the Job Agents necessary to its production and wait until all jobs are completed.

The Job Agent (figure 4) manages the route that its job follows inside the production system, by communicating with

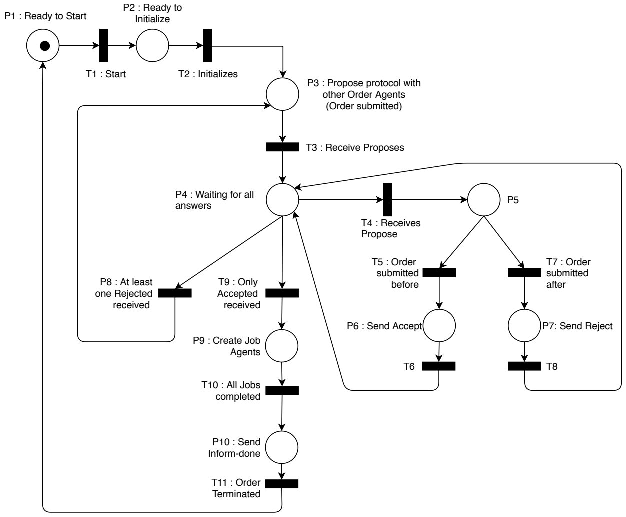

Fig. 3. Petri net behavioural model of the Order Agent

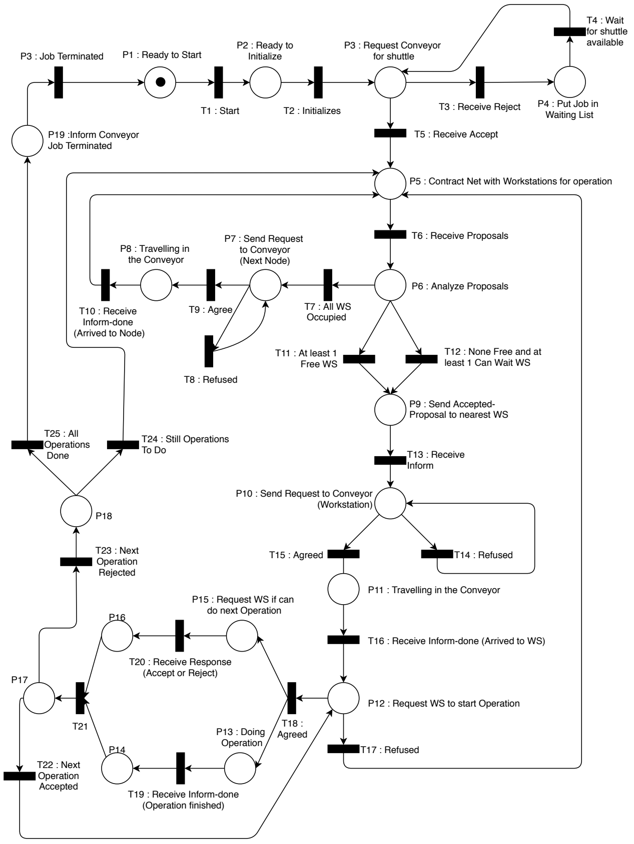

Fig. 4. Petri net behavioural model of the Job Agent

the Workstation Agents (figure 5) and Conveyor Agent (figure 6).

Workstations and Conveyor Agents are created at the beginning of the production. Workstations Agents register in the Directory Facilitator (DF) (yellow pages service of JADE) the operations they are capable to perform.

When Job Agent is created and initialized, its first action is to request a shuttle to Conveyor Agent, since there is a maximum number of shuttles available depending on the scenario under study. If there are shuttles available, job starts its production sequence. Otherwise, job is put on a waiting list until a shuttle is available.

In order to start the production sequence, Job Agent send a request to DF and starts the Contract-Net Protocol (CNP) with the workstations capable to perform the desired operation. Then, analyses the proposals of each workstation in order to choose the one most suitable to perform the operation. Depending on the status proposed by each workstation ("Free", "CanWait" or "Occupied"), the agent gives primacy to free workstations and among these choose the one nearer to its current location. If there are no free workstations, the agent use the same criteria for the workstations where the job can wait to be done. If all workstations are occupied or there are no workstations capable to perform the operation (in case of failure of the same), the agent send a request to conveyor agent to start a travel in the conveyor system while search for a free workstation.

At each node, job agent repeats the CNP with the workstations until a proposal is accepted or repeats the request to DF until a workstation became available.

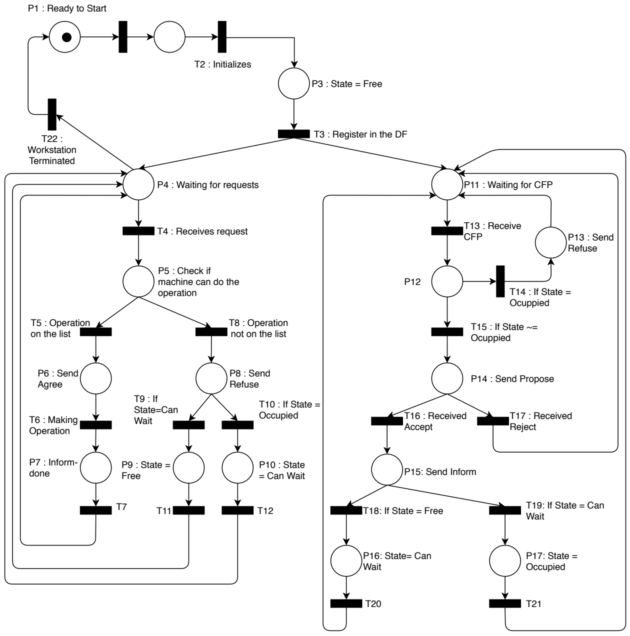

Fig. 5. Petri net behavioural model of the Workstation Agent

In the case a proposal is accepted, Job Agent send a request to Conveyor Agent asking for the shortest path until the desired workstation and initiates its travel in the conveyor. Arriving to the workstation, Job Agent send a request to start the operation. While the operation is being carried out, Job Agent also request if the workstation can perform the next operation in the sequence. If the Workstation Agent agrees, the next operation is performed at the end of the current operation. Otherwise, the job agent starts CNP with the workstations that have the capacity to perform the next desired operation.

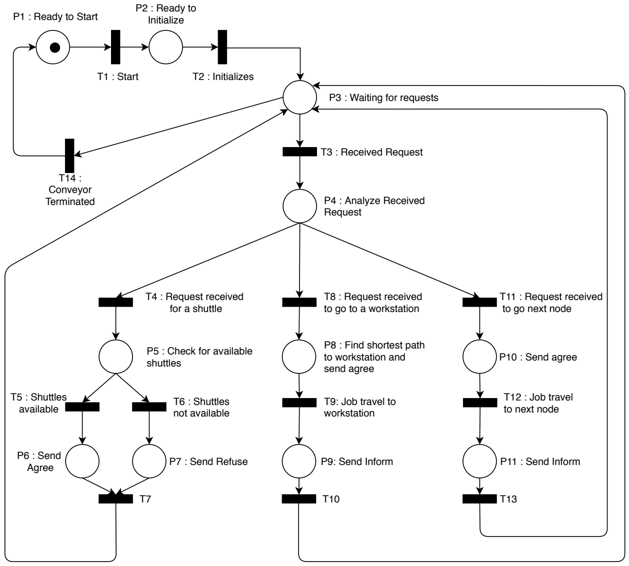

Fig. 6. Petri net behavioural model of the Conveyor Agent

After completing all the operations, the Job Agent inform the Conveyor Agent, so a shuttle is available to enter the conveyor system and perform the jobs in the waiting list.

As stated in II-D, this interaction between agents can both be described by petri nets but also through AUML. A short explanation of the production of the operation ”Axis” for job B is presented in figure 7.

Job Agent B requests DF for the workstations that can perform the operation ’Axis’ and initiate the CNP with those. Workstation 2 Agent proposes ’Free’ and Workstation 3 Agent proposes ’CanWait’. Job Agent B accepts the proposal from Workstation Agent 2 and initiates the request with Conveyor Agent. This one agrees, calculate the shortest path to workstation 2 and inform when Job Agent pass through nodes until reach the workstation. When arriving, Job Agent B request to perform the operation and Workstation Agent agrees, informing when it is done. Then two different scenarios are represent: At blue (behaviour 1), Job Agent request to perform the next operation in the sequence and Workstation Agent agree (since it is capable to do it).At red (behaviour 2), Workstation Agent rejects and Job Agent, at the end of the current operation, starts again the request to DF to perform the next desired operation as in the first step, following the same logic for the remaining operations.

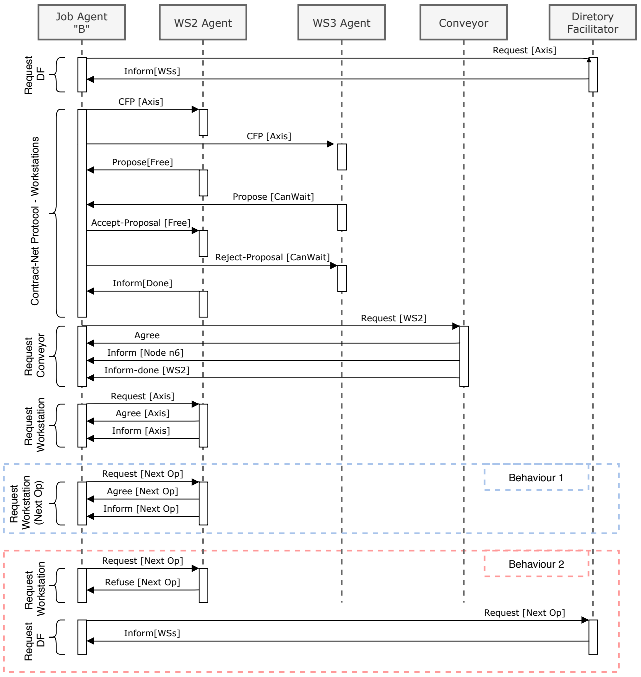

Fig. 7. AUML for Job Agent B performing operation ’Axis’

F. Decision Rules

Decision rules that manage the operation of the agents are a key aspect and consequently influence the global operation of the system. For that reason, jobs are governed by the following decision rules:

- Jobs are loaded by the order in which their orders are submitted
- Whenever is possible to load jobs in the manufacturing system, they are immediately loaded
- When analysing the proposals, jobs opt for workstation by the following priority: (1) Free workstations and (2) Nearest Workstation
- Jobs request all the operations that one workstation can provide before moving to another workstation

Some of those rules are briefly represented in both in petri nets and interaction diagrams.

III. SCENARIOS

The manufacturing control system was tested in different conditions defined in static and dynamic scenarios. A static scenario is a production scenario where all the data is known initially and there are no perturbations to be considered. However real life production systems are dynamic, facing perturbations that cannot be predicted in the beginning of the production. Considering this premises, a static scenario was chosen from the possible scenarios introduced in [9]. Taking in consideration the amount of data available for comparisons, scenario C0 was selected in conjunction with a set of dynamic scenarios (PS2, PS7, PS9 and PS10). These dynamic scenarios cause different disturbances in the static scenario thus testing different system capabilities. A summary of each scenario is presented in table III.

IV. EXPERIMENTAL RESULTS

Considering the flexible manufacturing system and that it is of utmost importance to have a clear view of the workstation

TABLE III

STATIC SCENARIO AND DYNAMIC SCENARIOS ASSOCIATED [9]

Static Scenario						
Code	No. Shuttles	Order #	BELT	AIP	LATE	Due Date
C0	4	#1 #2	1 -	- 1	- -	382 238

Dynamic Scenarios		
	Description	Parameters
PS2	At a given time, one of the machines was improved and is now able to perform a new kind of manufacturing operation which increases the flexibility level of the cell	**Machine:** M3. **Operation**[a]**:** L_comp **Start Time:** just after the departure of the second shuttle from M3 **Updated processing time:** 20 seconds
PS7	At a given time, a part of the conveyor system is due for maintenance in a given time window	**Start time:** Just after the fourth job is unloaded. The conveyor must no longer accept shuttles, and as soon as it empty, the maintenance starts **Duration**(seconds): 25 x Total number of jobs
PS9	At a given time, one of the redundant machines will go down in a given time window	**Machine:** M2. **Start time:** just after the departure of the first shuttle from M2. **Duration** (seconds): 25 x Total number of jobs
PS10	At a given time, one of the critical machines will go down in a given time window	**Machine:** M4. **Start time:** just after the departure of the second shuttle from M4. **Duration** (seconds): 25 x Total number of jobs

[a] Different operation from the original (I_comp) because original gives the same result for static scenario C0

allocation along the manufacturing process, gantt charts will be used to portray the experimental results along with the global makespan (C_{max}) used as a quantitive performance indicator.

For comparison purposes, the results obtained are compared both against simulated and real experimentations of a different approach called Potential Fields Approach (PFA). In PFA, machines emit attractive fields, depending on the operations that they can perform and availability, to attract the shuttles containing the jobs. Shuttles sense the fields through the cell and move to the most attractive one (see [15] for a detailed concept explanation). Static scenario is also compared with with Mixed-integer linear programming (MILP). These comparison results are reported in [9]

A. Static Scenario C0

The gantt chart portraying the operation of the system for static scenario C0 is presented in figure 8 and the makespan is 440s. As expected, there is an increase in the makespan (395s vs 440s) compared to the result from MILP presented in figure 9. However, analysing and comparing with PFA, the makespan is higher (448 vs 440) compared to MAS. This difference can be explained by some extra time present in PFA between loading and unloading jobs, since jobs' allocation in workstations are the same (see figure in [9]).

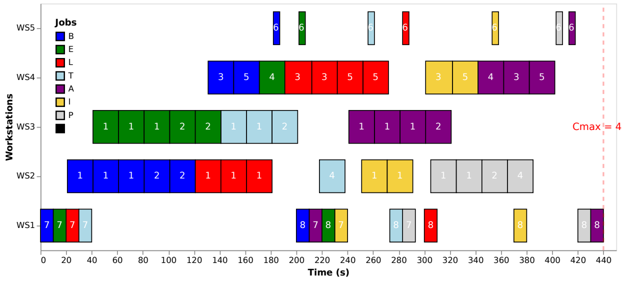

Fig. 8. Gantt chart Scenario C0 from MAS

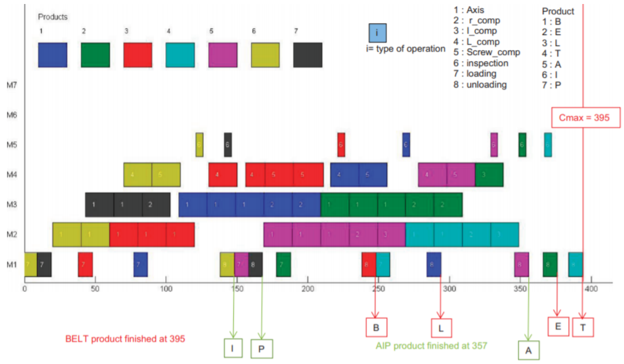

Fig. 9. Gantt chart Scenario C0 from MILP

B. Dynamic Scenario PS2

The gantt chart portraying the operation of the system for static scenario PS2 is presented in figure 10 and the makespan is 437s.

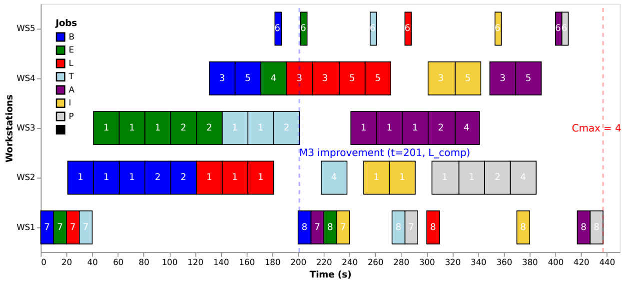

Fig. 10. Gantt chart Scenario PS2 from MAS

Second shuttle containing job T leaves worksation 3 ($t = 201$) and at this time machine 3 became capable to perform the operation L_comp besides the already mentioned operations in section II-A. This increase of flexibility in machine 3 results in a decrease of makespan (437s vs 440s) compared to static scenario C0. Job A performs operation L_comp in machine 3 instead of waiting to perform this operation in machine 4 and when arrive at worksation 4, it's already free to perform the necessary operations.

C. Dynamic Scenario PS7

The gantt chart portraying the operation of the system for static scenario PS7 is presented in figure 11 and the makespan is 456s.

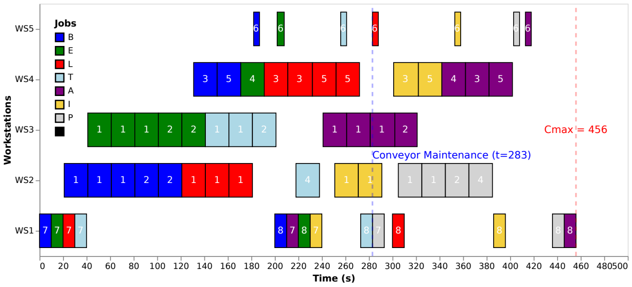

Fig. 11. Gantt chart Scenario PS7 from MAS

After fourth shuttle containing job T is unloaded ($t = 283$s), marked part in figure 1 fail and become inaccessible for shuttles. Conveyor no longer accept shuttles besides those already in production. In this scenario, it is possible to finish all orders, since all the jobs were in the conveyor system at the time conveyor part failed. However, it is possible to verify the adaptive capacity of the system with the increase of makespan in this scenario compared to static scenario C0 (456s vs 440s). This is mainly due to jobs changing the route from workstation 5 to workstation 1, which following the rules of shortest path would be through node 2 directly from workstation 5 but due to conveyor part failure was changed to pass in nodes 10, 11 and 1 before reaching node 2.

Compared to the result from PFA (456s vs 548s) presented in figure 12 there is an improve of 17%. Machine allocation is similar but, in this case, PFA is a real implementation which explains the difference due to real environment constraints.

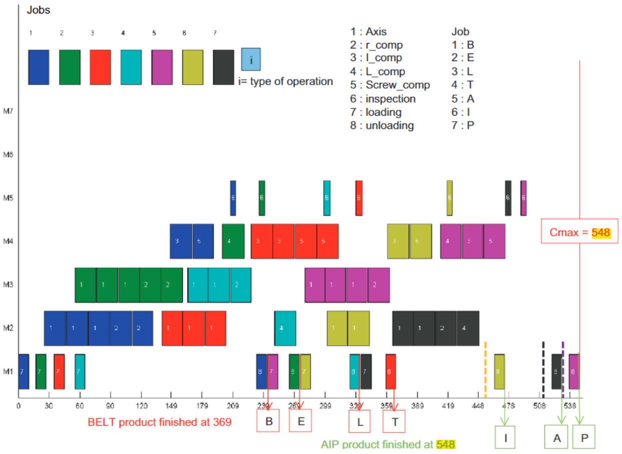

Fig. 12. Gantt chart Scenario PS7 from Potential fields approach

D. Dynamic Scenario PS9

The gantt chart portraying the operation of the system for dynamic scenario PS9 is presented in figure 13 and the makespan is 488s. A redudant machine (M2) goes down after the departure of the first shuttle containing job B. At this time, job T was waiting at the workstation 2 having been relocated to workstation 3 although only after it entered the conveyor system searching for a workstation since workstation 3 was the only available and was occupied at the time. Compared to PFA (figure 14), the makespan is slightly lower (488s vs 491s). Machines' allocation is the same in both approaches and again this difference can be explained by some extra time present in potential fields between loading and unloading jobs.

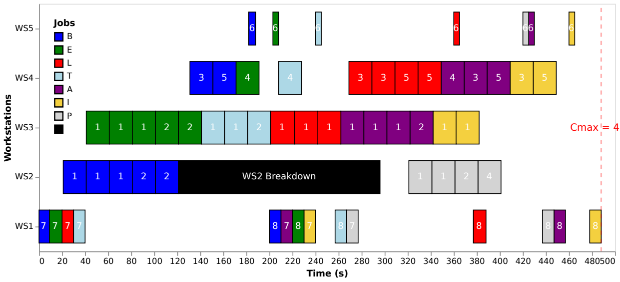

Fig. 13. Gantt chart Scenario PS9

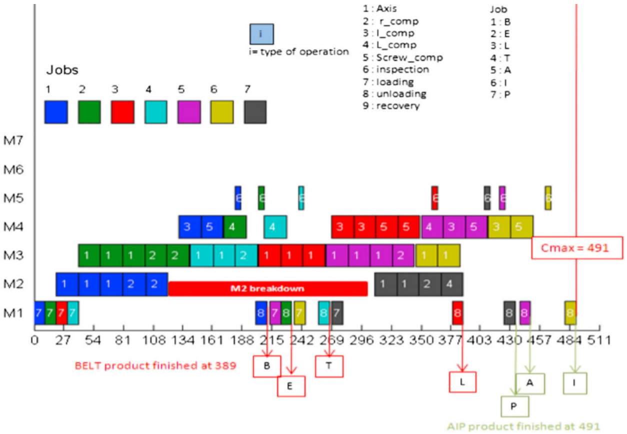

Fig. 14. Gantt chart Scenario PS9 from Potential fields approach

E. Dynamic Scenario PS10

The gantt chart portraying the operation of the system for dynamic scenario PS10 is presented in figure 15 and the makespan is 591s.

A critical machine (M4) goes down after the departure of the second shuttle containing job E. By this time, job L had already been accepted by workstation 4 to wait for the execution of operations and was on its way in conveyor system, so the searching process around conveyor begin. Job I also begin the searching process after conclusion in workstation 2, since workstation 4 were the only capable to proceed the following operation. When machine 4 became available, job L was at node 1 and job I was at node 11,

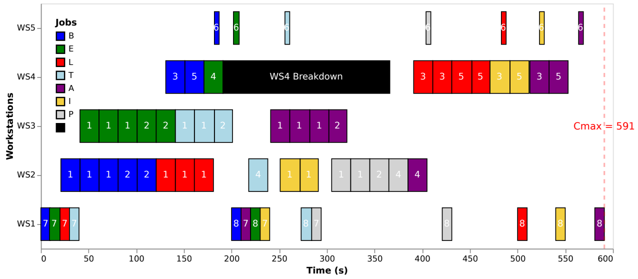

Fig. 15. Gantt chart Scenario PS10

so job L was accepted first since was closer from workstation 4 and Job I had to wait in the workstation.

The global makespan represents an increase of 34% in respect to the reference static scenario C0. As expected, the impact of a breakdown in one of the critical machines is considerably higher compared to a breakdown in one redundant machines (scenario PS9) in terms of makespan increase

V. CONCLUSIONS

The proposed multi-agent approach and respective comparisons made through the previous sections, namely with MILP and PFA, are summarized in table IV.

After testing the designed agent-based control system under static and dynamic scenarios, a lack of long term vision is observed in the distributed MAS control architecture, resulting in workstation allocations far from optimal, especially when compared with MILP in static scenario. Nevertheless, the capacity of the system to react and adapt to the failures induced in the dynamic scenarios reveals a promising solution for the control of flexible and reconfigurable systems that experience dynamic environments and require a very reactive behaviour.

On the other hand, the extremely simple behaviour of each agent in the MAS and the better results presented especially when compared with simulated PFA, can be a pronunciation for this approach usability in more complex manufacturing systems.

TABLE IV
RESULTS COMPARISON

Scenarios	**Multi-agent approach**	**MILP**	**Potential Fields**	**Potential Fields (Real Experiment)**
C0	440	395	448	528
PS2	437	-	-	-
PS7	456	-		548
PS9	488	-	491	589
PS10	591	-	-	-

VI. FUTURE WORK

Test and report the remaining static and dynamic scenarios present in the benchmark [9].

Provide agents with more information about the global system and reduce their lack of horizon creating an ontology to allow more complex interactions between agents and more exchanged information between workstations and jobs. Develop decision making algorithms to take advantage of the received system information and improve agents intelligence and the quality of autonomous decisions.

Design of an hybrid system combining a reative MAS and an optimization algorithm like MILP or a metaheuristic, allying an optimal static performance with the ability to react in case of dynamic disturbances. This could be accomplished by the creation of a supervisor agent in the MAS that would execute the optimization algorithm and advise other agents to act in accordance with its solution until a disturbance appears.

REFERENCES

[1] S. Wang, D. Li and C. Zhang. "Implementing Smart Factory of Indutrie 4.0: An Outlook". International Journal of Distributed Sensor Networks, 2016.

[2] P. Leitão. "Agent-based distributed manufacturing control: A state-of-the-art survey". Engineering Applications of Artificial Intelligence, 22(7):979–991, 2009.

[3] A. Colombo. "Industrial agents: towards collaborative production - automation, -management and -organization". IEEE Industrial Electronics Society Newsletter, 52((4)):17–18, 2005.

[4] P. Leitão, V. Marik, and P. Vrba. "Past, Present, and Future of Industrial Agent Applications". IEEE Transactions on Industrial Informatics, 9(4):2360–2372, Nov 2013.

[5] P. Leitão, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo. "Smart Agents in Industrial Cyber-Physical Systems". In Proceedings of the IEEE, volume 104, pages 1086–1101, 2016.

[6] L. Monostori, J. Vancza, and S. R. T. Kumara. "Agent-based systems for manufacturing". CIRP Annals - Manufacturing Technology, 55(2):697–720, Jan 2006.

[7] V. Marík and J. Lazanský. "Industrial applications of agent technologies". Control Engineering Practice, 15(11):1364–1380, 2007.

[8] P. Leitão and P. Vrba. "Recent Developments and Future Trends of Industrial Agents". HoloMAS 2011, LNAI 6867:15–28, 2011

[9] D. Trentesaux, C. Pach, A. Bekrar, Y. Sallez, T. Berger, T. Bonte, P. Leitão and J.Barbosa. "Benchmarking flexible job-shop scheduling and control systems". Control Engineeting Practice, vol. 21, no. 9, pp. 1204–1225, Sep. 2013.

[10] T. Coito, J. Viegas, M. Martins, M. Cunha, J. Figueiredo, S. Vieira and J. Sousa. "A Novel Framework for Intelligent Automation", IFAC-PapersOnline 52(13): 1825-30.

[11] J. Zhang, X. Yao, J. Zhou, J. Jiang, and X. Chen. "Self-Organizing Manufacturing: Current Status and Prospect for Industry 4.0". In Proceedings - 2017 5th International Conference on Enterprise Systems: Industrial Digitalization by Enterprise Systems, ES 2017, pages 319–326, 2017.

[12] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[13] FIPA Web Site: http://www.fipa.org/, accessed on January 15, 2020.

[14] B. Bauer, J. P. Muller, and J. Odell. Agent UML: A formalism for specifying multiagent software systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1957 LNCS(3):91–103, 2001

[15] N. Zbib, C. Pach, Y. Sallez, D.Trentesaux. "Heterarchical production control in manufacturing systems using the potential fields concept", Journal of Intelligent Manufacturing, 23(5), 1649-1670

[16] M. W. Khan, J. Wang, M. Ma, L. Xiong, P. Li, and F. Wu, "Optimal energy management and control aspects of distributed microgrid using multi-agent systems," Sustainable Cities and Society, vol. 44. Elsevier Ltd, pp. 855–870, 01-Jan-2019

[17] K. Raya-Díaz, C. Gaxiola-Pacheco, M. Castañón-Puga, L. Palafox, J. Castro, and D.-L. Flores, "Agent-Based Model for Automaticity Management of Traffic Flows across the Network," Applied Sciences, vol. 7, no. 9, p. 928, Sep. 2017