THE MODELLING OF MANUFACTURING SYSTEMS BY MEANS OF STATE MACHINES AND AGENTS

Krzysztof Foit

Silesian University of Technology, Department of Engineering Processes Automation and Integrated Manufacturing Systems, Konarskiego 18A St., 44-100, Gliwice, Poland

Corresponding author: Krzysztof Foit, krzysztof.foit@polsl.pl

Abstract: The changes taking place in industry, and in particular the implementation of the Industry 4.0 philosophy has significantly influenced the way of modelling of manufacturing processes. Because of the introduction of cyber-physical systems, the manufacturing systems are often modelled using multi-agent or holonic approach. Apart from the used methodology, the other important thing is the internal mechanism that determines how the model works. This paper attempts to describe these mechanisms using state machines and multi-agent approach. In this manner, using the signals responsible for switching the machine to a particular state, it becomes possible to show the processes that happen inside the agent as well as the communication between agents. For this purpose, a certain formalism changes are required in order to illustrate the communication between machine and its environment. As a result, not only a structural description of the system is presented, but also the way of its operation.

Key words: manufacturing system, Industry 4.0, modelling, agent, state machine

1. INTRODUCTION

The approach to manufacturing has radically changed in recent years. Implementing innovative solutions to make the production more efficient and simultaneously customer-oriented, requires the use of new modelling methods that go beyond purely technological issues. In the era of Industry 4.0, the concept of a factory takes on a different meaning, due to the decentralization of resources, creation of virtual enterprises or use of cloud solutions. The emergence of cyber-physical systems has also led to the concept of the digital twin that is an exact, virtual representation of a real system. In this sense, the machines involved in the production process share information not only with each other, but the communication is also carried out in the virtual environment, enabling a more efficient planning of the production process and use of resources.

In order to illustrate the operation of a technical means or communication between machines, phenomenological models are created that refer to the characterisation of the phenomenon as it is. Being focused on processes, the computer graphics models of machines and the other equipment play a background role, while the functional properties that have a direct impact on the technological process come to the fore. Following that rule, the model must describe the objects involved into process, including important properties, functions and processes (Tavola et al., 2017), as well as the signals responsible for communication and the control of the process flow. However, modern manufacturing systems are more ‘flexible’, ‘customer-oriented’ and ‘self-organizing’ what leads to the dynamic changes of system’s configuration. As a result, the philosophy of the model creation must be focused on the unit that maps a particular machine, rather than the whole system – as Tavola et al. point out, it is not the question how to model the entire structure, but what combination of components is the most suitable for the considered model. For example, creation of a model of self-organized systems structure is virtually impossible, because of its dynamic nature. Of course, it is possible to create a model of a single, chosen form, but many information is lost. The review of current literature shows that three, dominant ways are used in the field of modelling of such systems:

- using an object-oriented approach e.g. (Mize et al., 1992), (Anglani et al., 2002),
- using the multi-agent system approach e.g. (Unland, 2015), (Lima et al., 2006), (Shen, 2002), (Shen & Norrie, 1999),
- using the holonic approach, in conjunction with the methodology of agents e.g. (Van Brussel, 2014), (Botti & Giret, 2008), (Chirn & McFarlane, 2000), (Mella, 2009).

Components of the contemporary manufacturing system must be also equipped with the control system and communication interfaces in order to exchange information with the other parts of a system as well as to realize its tasks. Mentioned features are also the subjects of modelling and can be represented in different forms, like graphs, pseudocode, Petri Nets etc. while the last one is often used in research e.g. (Celaya et al., 2007), (Moldt & Wienberg, 1997), (Figat & Zieliński) in order to represent the model of a whole system or internal processes of system components.

This paper will present a different approach to the modelling of manufacturing systems, using agent-based approach and state machines formalism. The aim is to combine the flexibility of autonomous agent-based models with the representation of its internal control structure in the form of automata. The next part of this paper will briefly present the scientific background in reference to the topic of the paper. Then the assumptions of the new approach will be discussed. Finally, the simple example of application of the new method will be presented and the conclusions will close the paper.

2. SCIENTIFIC BACKGROUND, RESEARCH MOTIVATION

2.1 Scientific background

The general discussion about agents, its autonomy as well as the definition of an autonomous agent can be found for example in (Franklin & Graesser, 1997), (Wooldridge & Jennings, 1994), (Castelfranchi, 2010), (Luck & d'Inverno, 1995), (Luck et al., 1997) and many other. In turn, Unland makes the reference to the industry, defining an industrial agent (Unland, 2015). Being autonomous, an agent is often used in modelling of intelligent or adaptative manufacturing systems (Shen, 2002), (Shen & Norrie, 1999), (Lima et al., 2006). Such methods are also applicable during the process planning and supply chain management (Wong et al., 2006), (Lee & Kim, 2008). The Agent methodology is also widely used for modelling and analyzing of self-organizing systems. The major applications are connected with building the systems based on methods that mimic the processes taking place in nature, such as swarm or evolutionary algorithms (Leitão, 2012), (Barbosa et al., 2015), (Stypka et al. 2018).

The self-organization of manufacturing systems is also reflected in the representation of such systems in the form of a holarchy, i.e. structures composed of units called holons. The concept of the holon comes from philosophy (Koestler, 1967), (Koestler, 2013), and the main assumptions of this methodology are the basis of the so-called Holonic Manufacturing Systems (Botti & Giret, 2008), (Chirn & McFarlane, 2000), (Mella, 2009), (Van Brussel, 2014). In this case, the structure of the holon and the methods of modelled system representation in the form of a holon are much better systematized than in the case of agent-based systems (including the IEC 61449 standard or the HCBA method (Chirn & McFarlane, 2000)). However, some authors point to a strong relationship between the HMS and the agent-based representation of manufacturing system components, noting the complementarity of the agent-based and holonic approaches (Vrba et al., 2011), (Van Brussel, 2014), (Barbosa et al., 2015), (Suarez et al., 2013).

Referring to the description of the internal mechanisms of the agent (the 'heart', control system), some authors use appropriate languages, such as JADE, ACL, KQML TELESCRIPT, AGENT0 etc. (Bellifemine et al., 2001), (Chaib-draa & Dignum, 2002), (Parasumanna & Srinivasan, 2010), (Wooldridge & Jennings, 1994). In more recent publications, Petri Nets are used for the same purpose (Moldt &Wienberg, 1997), (Figat & Zieliński, 2019), (Celaya et al., 2007). The analysis of literature related to the subject matter of this paper has shown that the research conducted so far concerning the application of FSM in the field of manufacturing systems modelling is focused mainly on the field of robotics. Moreover, no reference is made to the agent-based modelling. Certain attempt to implement FSM for modelling of manufacturing systems can be found e.g. in (Doty & Van Aken, 1993), where the authors only mention the use of FSM for task sequencing, but do not give any details. In turn (Kube & Hong, 1997) describe the implementation of finite-state automata in quite a detailed way. The authors refer to the modelling of tasks in the environment of cooperating robots, and the use of FSM is limited to the description of tasks performed by the robot, assuming different levels of detail. In the mentioned publications there are no references to the object-based or agent-based model of the environment, but at the time of the publication the mentioned topics were not yet sufficiently developed.

More recent publications indicate that application of FSM is more or less the same, as mentioned above. The use of them mainly concern the programming of robot tasks (e.g. Herrero et al. 2017), (Wahrburg et al., 2015), but also a human-robot cooperation (Ding et al., 2013). The paper written by Smirnov et al. refers to the contemporary Internet of Thing devices and cyber-physical systems, represented by FSM and described by means of ontological modelling in order to achieve improvements in terms of programming and interoperability of these systems (Smirnov et al., 2020). Tavola et al. use the finite-state automata in modelling of production systems, but there is no reference to the agent-based model (Tavola et al., 2017).

A deeper analysis of the scientific publication databases leads to the conclusion that the FSA methodology is used in conjunction with agent-based models, but in majority is not related to the manufacturing processes (Naumov & Shalyto, 2003), (Shalyto et al., 2005), (Yartsev et al., 2005). There are few publications that deal with this issue in the industrial context, e.g. (Luo et al., 2013), (El Mouayni et al. 2017), but the presented approaches are still in the development phase.

2.2 Research motivation

The analysis of the literature related to the subject matter of this paper has shown that relatively small number of studies relate to modelling of manufacturing systems, taking into account both the agent approach and the description of the agent in the form of finite-state machines. In the analysed works, such description is given in general terms, without going into the details of the process being carried out. Hence, there are still open problems, concerning in particular the agent's communication with its environment, i.e. other machines involved in the production process. This implies a justified need for further research on such issues.

3. FSM AS A DESCRIPTION OF AGENT'S BEHAVOIUR

3.1 Agent as a model of manufacturing system components

An agent is an entity that acts in some kind of environment, perceive it and communicate with other agents. Being autonomous it also has ability to perform actions that are needed to achieve given goals. Using the simplest model of an agent, we may distinguish three sets. Let's denote the set of actions by *A*, set of agent's internal states by *S*, set of signals (information) coming from the environment by I and set of percepts by *P*. In this way it is possible to define internal functions as

$$f_1 : P \rightarrow S \quad (1)$$

$$f_2 : S \rightarrow A \quad (2)$$

Of course action sets the state of environment $e \in E$ that is again detected by the agent.

$$f_3 : A \rightarrow E \quad (3)$$

$$f_4 : E \rightarrow I \quad (4)$$

The last one is an internal function that is responsible for change the internal state of agent according to the result of performed action

$$f_5 : S \times A \rightarrow S \quad (5)$$

In the Figure 1 the simplified structure of agent is shown, along with the circulation of information and functions presented in equations (1)-(5). The perception block is marked with *P* and is responsible for transforming input signal into percepts. *S* is the state block that will be discussed later. *A* represents the action block that controls the actuators.

Referring to the diagram shown in Figure 1, it is possible to assign appropriate blocks from the diagram to separate sub-systems of the real machine (e.g. robot, CNC lathe etc). In this way, the perception block can be related to sensors and the input of communication devices, the state block to the control system, and the action block to the actuating system and output of communication devices.

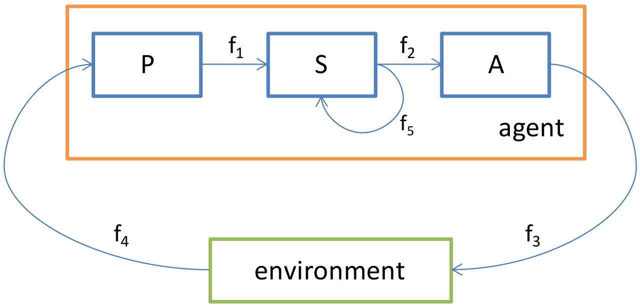

Fig. 1. Simplified model of an agent acting in specific environment

It should be noted that the output of an action block does not have to be of a physical nature – it could also be a certain signal (e.g. electric, pneumatic, light). Similarly, the term 'environment' can be understood as a limited part of reality, which is the domain of agent's activity.

3.2 State machines as the representation of the process

State machine, called also finite-state machine or finite-state automaton, is an abstract, mathematical model of computation. On the other hand, it is also often used for behaviour modelling. Because of strong relationship with mathematics, they are widely used in computer science, also in connection with the development of artificial intelligence.

Finite-state machines are divided into two main groups:

- Deterministic Finite-state Automata (DFA) – where each state has exactly one transition for each possible input,
- Non-deterministic Finite-state Automata (NFA) – where an input can lead to more than one or no transition.

Further subdivision of the FSM will not be discussed in this paper – detailed description is available in the relevant literature.

The deterministic finite-state machine M is one of the automata classes and is defined by initial state $q_0 \in Q$, finite sets of states *Q*, finite set of symbols (alphabet) *Σ*, transition functions *δ* and final state (or set of final states) *F* (equation 6).

$$M = \{Q, \Sigma, \delta, q_0, F\} \qquad (6)$$

The simplest example of FSM is a lamp with a button. The initial state is OFF, and pressing the button turns the lamp on (changes the machine state to ON). Subsequent presses switch the lamp on and off (change the machine state between the ON and OFF) (Figure 2).

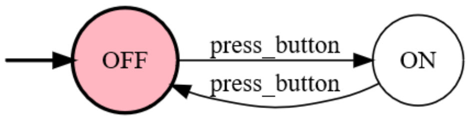

Fig. 2. The state diagram of very simple FSM

The state diagram shown in Figure 2 is a graphical representation of a finite-state automaton. It consists of states (represented by circles) and transitions (represented by arcs). Other form of representation is provided by state transition tables, but not be considered there.

3.3 FSM as a representation of agent's internal states

Concerning the structure of an agent, shown in Figure 1, the state block *S* could be replaced by embedded state machine *M* (Figure 3). In order to carry out this operation, first of all the input alphabet must be defined as

$$\Sigma = \Sigma_P \cup \Sigma_I \qquad (7)$$

where Σ_P is a subset of alphabet based on precepts, and Σ_I is a subset of alphabet based on the internal signals that change the agent's state according to the result of performed action (internal feedback). The set of agent's states *S* is equipollent with the set of FSM set of states *Q*. Functions f_1 and f_5 are realized internally by automaton. Function f_2 is still needed, but can be omitted when automaton has also an output alphabet and gives output values.

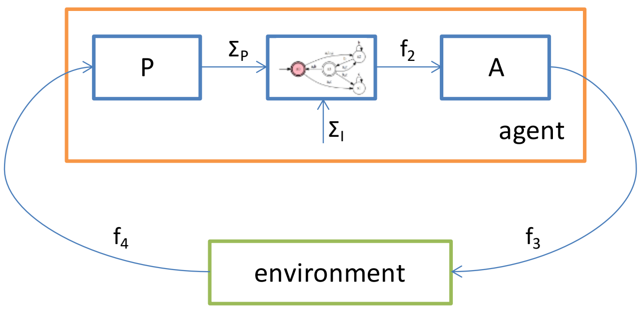

Fig. 3. FSM replacing the state block *S* in internal structure of an agent

The description of the agent's internal processes using FSM does not require any changes to the formalism. Most often, however, we deal with a multi-agent environment in which the correct realisation of the process is guaranteed by the use of appropriate communication between agents.

3.4 The multi-agent environment and FSM-based description of internal processes

In the multi-agent environment, the communications between agents plays important role during the process of realisation of the common aim. The message passing could use sophisticated channels and protocols, but the information could be also passed between agents indirectly, by changing the environment and sense these changes.

In the industry different kinds of sensors are used for gathering the information about the environment. Such information can be useful not only for particular agent (machine, device), but also for two or more agents. For example, a group of robots should wait during the operation of conveyor – the common signal is useful for several machines. During the modelling the signal from the sensor is represented by percept that in turn is a part of alphabet of the FSM, which replaces the state block in the structure of an agent.

The production process has a sequential nature. The individual tasks are distributed among the machines in a way that guarantees that the final product will be in the correct form. Therefore, it can be said that the manufacturing process is a discrete process in which the sequence of activities is clearly defined. The way the machines work can be described by means of a sequence of consecutive activities. By using the multi-agent approach, it is possible to illustrate the cooperation and communication between the components of the industrial environment. To describe processes taking place inside the unit (agent), finite-state machines, represented in the form of state diagrams, were used. Apart from the standard set of symbols, in addition two other signs were defined, which are shown in Figure 4. They refer to the states and has a special meaning. The first one – a subroutine – is used for grouping activities, which are performed by a given unit, the detailed consideration of which does not bring anything new to the analysis of a given case. For example, it may be a process of manipulating an object by a robot, which is performed according to a certain internal program and based on internal feedback.

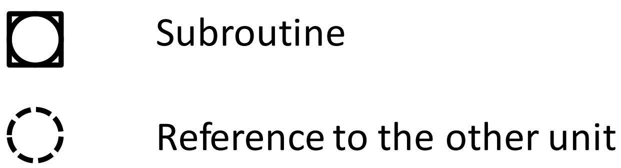

Fig. 4. Additional symbols introduced into state diagrams

The second symbol, indicating the state related to the passing of control to the another unit (machine), requires a short comment. Formally, the use of a regular symbol for such state could suggest that we are dealing with a non-deterministic state machine and therefore the new symbol is necessary. From the model point of view, it describes 'delegation' of the control to another unit in order to realize another part of the process. The last state of 'delegating' FSM is preserved. In such way it is easier to analyse the diagrams, because of the coherent description of operations performed by different units or organisational groups. Both symbols are not used in FSM simulators, so their use is limited only to handwritten diagrams – in this case they may facilitate the analysis of the system.

4. THE EXAMPLE OF APPLICATION

4.1 Description of the analysed system

Let's consider the part of manufacturing system, a cell that consists of the conveyor, two robots, lathe, milling machine and two storages (Figure 5). The part is picked by robot *R1* from the input buffer and then machined using lathe *L*. Next the robot *R1* places the part on the conveyor *C*. The object is moved until the sensor *sens_1* detects the part and stops the conveyor. Then the robot *R2* pick the object and move it to the milling machine *M*. After machining, the product is placed in the output buffer by the robot *R2*.

In order to illustrate the process, five cooperating agents will be considered: the conveyor, two machines and two robots. In order to simplify the analysis, other, supporting types of agents, will not be considered. The internal structure of agents is identical to that presented in Figure 3.

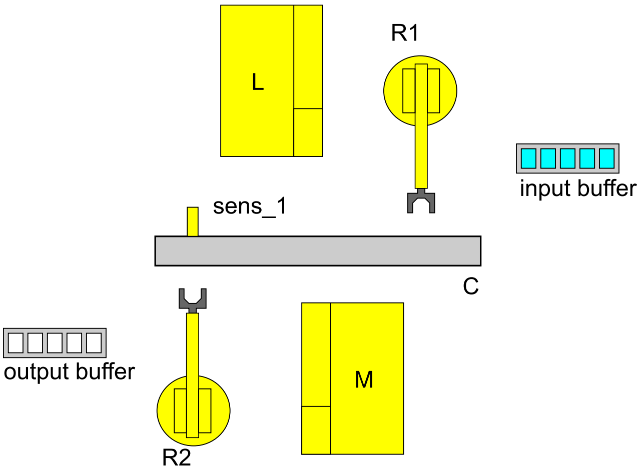

Fig. 5. The considered system

The Figures 6-10 show the internal FMS of individual agents, determining their behaviour. The alphabet and states are presented in a form that is easy to understand - symbolic markings have been dropped and abbreviations have been used instead. Names ending with a '#' are thus highlighted as generated by the action block and passed on to another agent. The explanation of the selected abbreviations is summarised in Table 1.

Table 1. Explanation of selected abbreviations used in Figures 6-10

States	Meaning
door_opened / door_closed	state of the machine door
full / empty	Full when the part is inside the machine, empty if not
fixed / released	State of machine chuck/clamp
inp_buf / out_buf	The robot's gripper is positioned above the input (output) buffer and ready to pick (place) a part
grip_opened / grip_closed	State of the robot's gripper
picked	Part is picked by the robot and ready to move
mill / lathe / conveyor	The robot's gripper is positioned near the milling machine, lathe or conveyor
mount_pos	The robot gripper is positioned near the machine's chunk/clamp and is ready to put/get the part into it
Alphabet	Meaning
load / unload	Starts the procedure on loading/unloading the part to/from the machine
move_M / move_L / move_C / move_out_b	Triggers the movement of the manipulator to the positions near the milling machine, lathe, conveyor, output buffer
start_C	Starts the conveyor
sens_1	Part is detected by sensor 1
open / close	Triggers the machine door

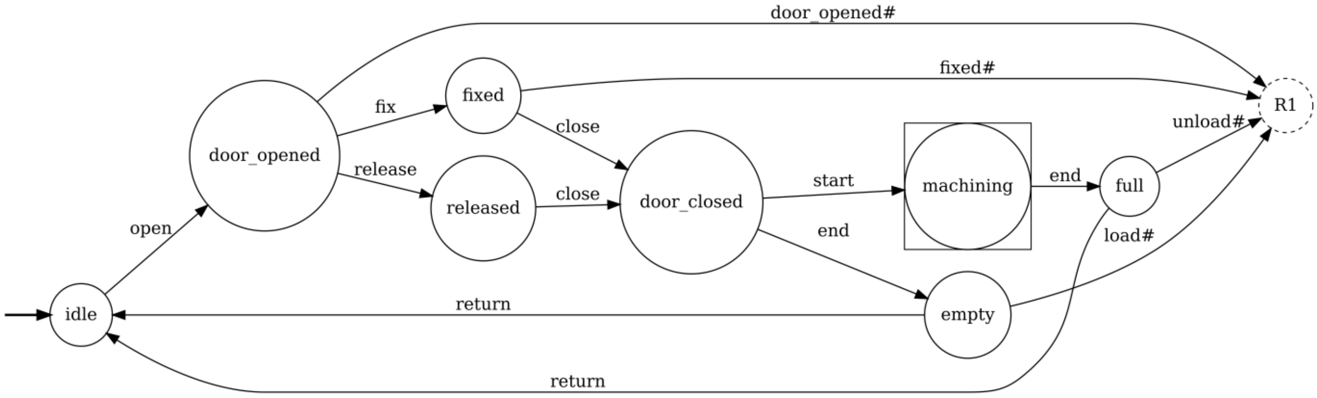

Fig. 6. The state diagram of internal FMS of lathe agent

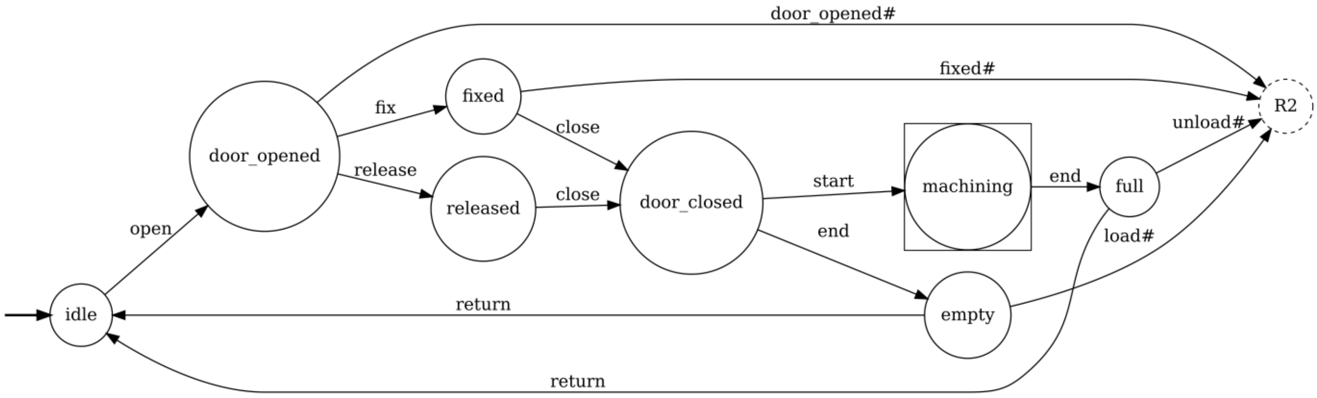

Fig. 7. The state diagram of internal FMS of milling machine agent

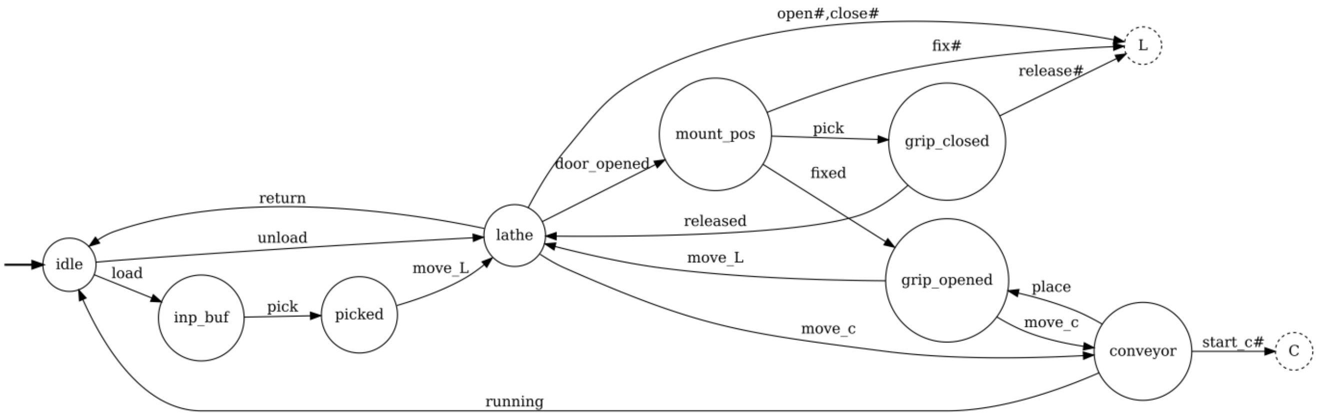

Fig. 8. The state diagram of internal FMS of Robot 1 agent

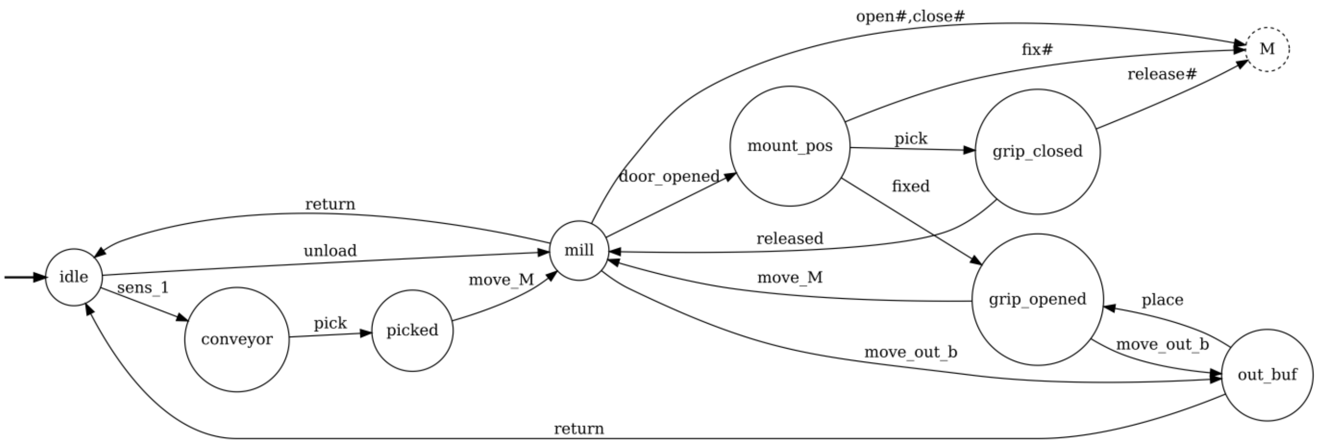

Fig. 9. The state diagram of internal FMS of Robot 2 agent

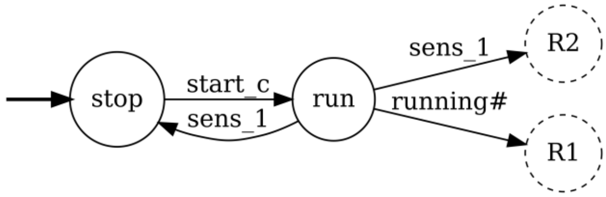

Fig. 10. The state diagram of internal FMS of conveyor agent

4.2 Discussion

As it can be seen, the use of graphical representation of finite-state machines in the form of state diagrams could be a good illustration of processes that occurs inside of the entity – in this case, inside of the agent. Also the state relations are presented in a readable manner.

In this example, five finite-state automata have been implemented – one per each machine involved in the manufacturing process. Moreover, it has been assumed that every machine will be represented as an agent. In this way, a multi-agent system has been obtained, excluding the administrative, supervisory, communication etc. agents. A finite-state machine has been embedded into the state block of agent and its purpose is to represent the internal states of a given agent. As a result, similar structures have been obtained for similar machines, in which the differences come down to the specific states, while the state diagrams themselves look similar from the graphic side. It seems obvious, but at the stage of creating individual machines, it gives the opportunity to use some kind of template. Thus, the identical internal structure of each agent allows to adopt a certain general pattern of behaviour for creating a model that could also be used for agents proper to management layer.

A limitation of the proposed form of systems' representation is the inability to carry out more than one threads simultaneously. For example, if a CNC lathe will perform the turning and drilling process simultaneously, the presented approach will not allow to describe this process in the form of one finite-state machine. In this case, it is necessary to look for other solutions. At this stage of development, the possible solution is to generalize the whole process by introducing a subroutine symbol. However, the number of threads executed simultaneously by agent is not specified, so further research may consider the parallel operation of several finite-state machines at the same time implemented in the same agent.

Two new elements were also added to the finite-state machine formalism, which are used in the graphical representation of the automaton in the form of a state diagram. The first one is the designation of the subroutine that allows to generalise a certain fragment of the operation of the entire machine. The example presented above, concerning the lathe and turning, is a good explanation of the subroutine meaning. In most cases, it is enough to know that the machine tool is in the state of executing a certain process, without going into details of its activity. The second graphic sign is the symbol of transferring control to the another unit. Drawn similarly to the state mark, for given automaton it is in fact a state of waiting for the other unit to finish its work and pass the control back to the calling FSM. Seemingly it looks similar to the subroutine, but there is the significant difference: the subroutine is a part executed within the same machine, while the passing of the control is carried out between two machines. Such 'switching' between agents is also a way of communication, allowing agents to cooperate with each other.

In order to compare the presented approach with the other similar methods, the Petri Nets comes to the fore. Regarding to the modelling of different types of manufacturing systems, a majority of publications have been dedicated to the use of Petri Nets in comparison with the low interest in the implementation of finite-state machines. This inequality could be a kind of challenge and motivation for research, but on the other hand the lack of relevant publications hinders critical approach to the results.

In comparison to the Petri Nets, finite-state machines seem to be more suitable for representing the internal behaviour of agents because could be easily described using state diagrams. Thus the principle of operation of the agent may be simulated even without any software, using only a sheet of paper and a counter – like playing a board game. In many cases creation and testing of algorithms could be done in a very simple way. Using special simulation software, a finite-state machine could be also converted to the code written in high level computer programming languages, like Java or C. On the other hand, Petri Nets have a well developed formalism with many forks and extensions. The natural representation of Petri Net is a graphic form that in turn could be less understandable without comments in comparison to FSM. The significant advantage of Petri Nets is the ability of processing more than one thread at once.

To sum up it must be said that both Petri Nets and FSM are useful during the process of modelling the internal processes of agents with remark that FSM are more intuitive form of presentation.

5. CONCLUSIONS

This paper has presented the foundations of a new approach to modelling of manufacturing systems, based on a multi-agent system with the use of elements of finite-state machines formalism. The state diagrams were used to describe the internal mechanisms of agents' operation – in this case the machines taking part in the manufacturing process. The problem of communication between units has also been addressed. For this purpose, an additional symbol has been introduced in the state diagrams that unambiguously identify the signal and the unit to what it is transmitted. Moreover, the second new symbol identifies a group of states or a subroutine. This allows to adjust the depth of presentation during the creation of a state diagram, hiding the unnecessary details. The example of application of the new approach has been also presented, with particular emphasis on the information exchange between units. Taking into account the holarchy of agents, as one of the possible organizational structures, the presented method allows describing the communication between the units on the same level, as well as the communication between levels. Moreover, using the common model of agent's internal structure, it is possible to create units that have different internal behaviour.

One of the most important problems related to the further development of the method is the attempt to create simulation software, taking into account the possibility of communication between the simulated units. Currently it is possible to simulate single units using finite-state machines simulation software, what can help during the validation of the model. The second disadvantage that must be concerned during the further research is a problem with the simulation of concurrent threads, because such kind of processing is often used in contemporary machine tools.

6. REFERENCES

1. Anglani, A., Grieco, A., Pacella, M., Tolio, T., (2002), *Object-oriented modeling and simulation of flexible manufacturing systems: a rule-based procedure*, Simulation Modelling Practice and Theory, **10**(3), pp. 209–234, https://doi.org/10/cxwgrt2.

2. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., (2015), *Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution*, Computers in Industry, **66**, pp. 99–111. https://doi.org/10.1016/j.compind.2014.10.011.

3.Bellifemine, F., Poggi, A., Rimassa, G., (2001), *Developing Multi-Agent Systems with JADE*, Intelligent Agents VII Agent Theories Architectures and Languages, Lecture Notes in Computer Science, Castelfranchi, C., Lespérance, Y. (Eds.), Springer, Berlin, Heidelberg, pp. 89–103. https://doi.org/10.1007/3-540-44631-1_7.

4. Botti, V., and Giret, A., (eds), (2008), *Holonic Manufacturing Systems*, ANEMONA: A Mulit-agent Methodology for Holonic Manufacturing Systems. London: Springer London, pp. 7–20. doi: 10.1007/978-1-84800-310-1_2.

5. Castelfranchi, C., (2010), *Bye-Bye Agents? Not*, IEEE Internet Computing, **14**(2), pp. 93–96. doi: 10/bz3zxn.

6. Celaya, J.R., Desrochers, A.A., Graves, R.J., (2007), *Modeling and analysis of multi-agent systems using petri nets*, 2007 IEEE International Conference on Systems, Man and Cybernetics, pp. 1439–1444. https://doi.org/10.1109/ICSMC.2007.4413960.

7. Chaib-draa, B., Dignum, F., (2002), *Trends in Agent Communication Language*, Computational Intelligence, **18**(2), pp. 89–101. https://doi.org/10.1111/1467-8640.00184.

8. Chirn, J-L., McFarlane, D. C. (2000) *A holonic component-based approach to reconfigurable manufacturing control architecture*, Proceedings 11th International Workshop on Database and Expert Systems Applications, 11th International Workshop on Database and Expert Systems Applications, London, UK: IEEE Comput. Soc, pp. 219–223. doi: 10.1109/DEXA.2000.875030.

9. Ding, H., Heyn, J., Matthias, B., Staab, H., (2013), *Structured collaborative behavior of industrial robots in mixed human-robot environments*, 2013 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1101–1106. https://doi.org/10.1109/CoASE.2013.6653962.

10. Doty, K. L., Van Aken, R. E., (1993), *Swarm robot materials handling paradigm for a manufacturing workcell*, Proceedings IEEE International Conference on Robotics and Automation, [1993] IEEE International Conference on Robotics and Automation, Atlanta, GA, USA: IEEE Comput. Soc. Press, pp. 778–782. doi: 10.1109/ROBOT.1993.292072.

11. El Mouayni, I., Etienne, A., Siadat, A., Dantan, J.-Y., Lux, A., (2017), *AEN-PRO: Agent-based simulation tool for performance and working conditions assessment in production systems using workers' margins of manoeuver*, 20th IFAC World Congress, IFAC-PapersOnLine **50**(1), pp. 14236–14241. https://doi.org/10.1016/j.ifacol.2017.08.2102.

12. Figat, M., Zieliński, C., (2019), *Methodology of Designing Multi-Agent Robot Control Systems Utilising Hierarchical Petri Nets*, 2019 International Conference on Robotics and Automation (ICRA), pp. 3363–3369, https://doi.org/10.1109/ICRA.2019.8794201.

13. Franklin, S. and Graesser, A., (1997), *Is It an*

agent, or just a program? A taxonomy for autonomous agents, Intelligent Agents III Agent Theories, Architectures, and Languages. Müller, J. P., Wooldridge, M. J., and Jennings, N. R. (eds), pp. 21–35, Berlin, Heidelberg: Springer Berlin Heidelberg.
14. Herrero, H., Moughlbay, A. A., Outón, J. L., Sallé, D., de Ipiña, K. L., (2017), *Skill based robot programming: Assembly, vision and Workspace Monitoring skill interaction*, Bioinspired Intelligence for Machine Learning, **255**, pp. 61–70. https://doi.org/10.1016/j.neucom.2016.09.133.
15. Koestler, A., (1967), *The Ghost in the Machine*, Macmillan. Available at: https://books.google.pl/books?id=GobyYUN_Jl8C.
16. Koestler, A., (2013), *Beyond Atomism and Holism - The Concept of the Holon*, The Rules of The Game: Cross-disciplinary Essays on Models in Scholarly Thought, Shanin, T. (ed.), pp. 233–247, Routledge, London.
17. Kube, C.R., Hong, Z., (1997), *Task Modelling in Collective Robotics*, Autonomous Robots, **4**(1), pp. 53–72, https://doi.org/10.1023/A:1008859119831.
18. Lee, J.H., Kim, C.O., (2008), *Multi-agent systems applications in manufacturing systems and supply chain management: a review paper*, International Journal of Production Research, **46**(1), pp. 233–265. https://doi.org/10.1080/00207540701441921.
19 Leitão, P., Barbosa, J., Trentesaux, D., (2012), *Bio-inspired multi-agent systems for reconfigurable manufacturing systems*, Engineering Applications of Artificial Intelligence, **25**(5), pp. 934–944. https://doi.org/10.1016/j.engappai.2011.09.025.
20. Lima, R. M., Sousa, R. M., Martins, P. J., (2006), *Distributed production planning and control agent-based system*, International Journal of Production Research, **44**(18–19), 3693–3709. https://doi.org/10/bmb6kv
21. Luck, M. and d'Inverno, M., (1995), *A Formal Framework for Agency and Autonomy*, ICMAS, pp. 254–260.
22. Luck, M., Griffiths, N., d'Inverno, M., (1997), *From agent theory to agent construction: A case study*, in Müller, J. P., Wooldridge, M. J., and Jennings, N. R. (eds) Intelligent Agents III Agent Theories, Architectures, and Languages. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 49–63.
23. Luo, S., Luo, G., Zhao, X., (2013), *Common production process modeling for MES based on multi-agent*, 2013 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), IEEE, Bangkok, Thailand, pp. 1582–1586. https://doi.org/10.1109/IEEM.2013.6962676.
24. Mella, P., (2009), *The Holonic Revolution, Holons, Holarchies and Holonic Networks. The Ghost in the Production Machine*, Pavia University Press, Available from https://doi.org/10.13140/2.1.1954.5922, Accessed: 01/07/2019.
25. Mize, J.H., Bhuskute, H.C., Pratt, D.B., Kamath, M., (1992), *Modeling Of Integrated Manufacturing Systems Using An Object-Oriented Approach*, IIE Transactions, **24**(3), pp. 14–26, https://doi.org/10/b43vmw.
26. Moldt, D., Wienberg, F., (1997), *Multi-agent-systems based on coloured Petri nets*, Azéma, P., Balbo, G. (Eds.), Application and Theory of Petri Nets, (1997), Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 82–101, https://doi.org/10.1007/3-540-63139-9_31.
27. Naumov, L., Shalyto, A., (2003), *Automata theory for multi-agent systems implementation*, IEMC '03 Proceedings, Managing Technologically Driven Organizations: The Human Side of Innovation and Change (IEEE Cat. No.03CH37502), pp. 65–70. https://doi.org/10.1109/KIMAS.2003.1245023.
28. Parasumanna Gokulan, B., Srinivasan, D., (2010), *An Introduction to Multi-Agent Systems*, Studies in Computational Intelligence, pp. 1–27. doi: 10.1007/978-3-642-14435-6_1.
29. Shalyto, A., Naumov, L., Korneev, G., (2005), *Methods of object-oriented reactive agents implementation on the basis of finite automata*, International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2005, pp. 460–465, https://doi.org/10.1109/KIMAS.2005.1427125.
30. Shen, W., (2002), *Distributed Manufacturing Scheduling Using Intelligent Agents*, IEEE Intelligent Systems, **17**(1), pp. 88–94.
31. Shen, W., Norrie, D. H., (1999), *Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey*, Knowledge and Information Systems, **1**(2), pp. 129–156. doi: 10/ggn85j.
32. Smirnov, A, Shilov, N., Shchekotov, M., (2020), *Ontology-Based Modelling of State Machines for Production Robots in Smart Manufacturing Systems*, International Journal of Embedded and Real-Time Communication Systems (IJERTCS). Hershey, PA, USA: IGI Global, **11**(2), pp. 76–91, doi: 10.4018/IJERTCS.2020040105.
33. Suárez, S., Leitao, P., Adam, E., (2013), *Holonic recursiveness with multi-agent system technologies*, Trends in Practical Applications of Agents and Multiagent Systems, Advances in Intelligent Systems and Computing, 221, Perez, J. et al. (eds.), pp. 103-111, Springer, pp. 103–111, Springer, Cham. https://doi.org/10.1007/978-3-319-00563-8_13.
34. Stypka, J., Turek, W., Byrski, A., Kisiel-Dorohinicki, M., Barwell, A.D., Brown, C., Hammond, K., Janjic, V., (2018), *The Missing Link! A New Skeleton for Evolutionary Multi-Agent Systems in Erlang*, International Journal of Parallel Programming, **46**(1), pp. 4–22, https://doi.org/10.1007/s10766-017-0503-4
35. Tavola, G., Taisch, M., Boschi, F., (2017), *A*

standard approach to production systems modelling based on Finite-state Automata, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). Presented at the 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), IEEE, Emden, pp. 1117–1122. https://doi.org/10.1109/INDIN.2017.8104930.
36. Unland, R., (2015), *Chapter 2 - Industrial Agents*, Industrial Agents: Emerging Applications of Software Agents in Industry, Leitão, P. and Karnouskos, S. (eds), pp. 23–44, Boston: Morgan Kaufmann, doi: 10.1016/B978-0-12-800341-1.00002-4.
37. Van Brussel, H., (2014), *Holonic Manufacturing Systems*, CIRP Encyclopedia of Production Engineering, W L. Laperrière & G. Reinhart (ed.), pp. 654–659, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20617-7_6556.
38. Vrba, P., Tichý, P., Mařík, V., Hall, K. H., Staron, R. J., Maturana, F. P., Kadera, P., (2011), *Rockwell Automation's Holonic and Multiagent Control Systems Compendium*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), **41**(1), pp. 14–30. https://doi.org/10/b93s6w.
39. Wahrburg, A., Zeiss, S., Matthias, B., Peters, J., Ding, H., (2015), *Combined pose-wrench and state machine representation for modeling Robotic Assembly Skills*, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 852–857. https://doi.org/10.1109/ IROS.2015.7353471.
40. Wooldridge, M. J., Jennings, N. R., (1994), *Agent Theories, Architectures and Languages: A Survey*, ECAI94 Workshop on Agent Theories Architectures and Languages (01/01/94), Woolridge, M. J. and Jennings, N. R. (eds), pp. 1–32, Available at: https://eprints.soton.ac.uk/252177/ (Accessed: 8 June 2020).
41. Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K., (2006), *Dynamic shop floor scheduling in multi-agent manufacturing systems*, Expert Systems with Applications, **31**(3), pp. 486–494. https://doi.org/10.1016/j.eswa.2005.09.073.
42. Yartsev, B., Korneev, G., Shalyto, A., Kotov, V., (2005), *Automata-based programming of the reactive multi-agent control systems*, International Conference on Integration of Knowledge Intensive Multi-Agent Systems, pp. 449–453. https://doi.org/10.1109/ KIMAS.2005.1427123.

Received: July 17, 2020 / Accepted: December 20, 2020 / Paper available online: December 25, 2020 © International Journal of Modern Manufacturing Technologies