IOP Conference Series: Materials
Science and Engineering

Object as a model of intelligent robot in the virtual workspace

To cite this article: K Foit *et al* 2015 *IOP Conf. Ser.: Mater. Sci. Eng.* **95** 012108

IOP Conf. Series: Materials Science and Engineering **95** (2015) 012108 doi:10.1088/1757-899X/95/1/012108

Object as a model of intelligent robot in the virtual workspace

K Foit[1], A Gwiazda[1], W Banas[1], A Sekala[1] and P Hryniewicz[1]
[1]Silesian University of Technology, The Faculty Of Mechanical Engineering, Institute Of Engineering Processes Automation And Integrated Manufacturing Systems, ul. Konarskiego 18a 44-100 Gliwice, Poland

E-mail: krzysztof.foit@polsl.pl

Abstract. The contemporary industry requires that every element of a production line will fit into the global schema, which is connected with the global structure of business. There is the need to find the practical and effective ways of the design and management of the production process. The term "effective" should be understood in a manner that there exists a method, which allows building a system of nodes and relations in order to describe the role of the particular machine in the production process. Among all the machines involved in the manufacturing process, industrial robots are the most complex ones. This complexity is reflected in the realization of elaborated tasks, involving handling, transporting or orienting the objects in a work space, and even performing simple machining processes, such as deburring, grinding, painting, applying adhesives and sealants etc. The robot also performs some activities connected with automatic tool changing and operating the equipment mounted on the wrist of the robot. Because of having the programmable control system, the robot also performs additional activities connected with sensors, vision systems, operating the storages of manipulated objects, tools or grippers, measuring stands, etc. For this reason the description of the robot as a part of production system should take into account the specific nature of this machine: the robot is a substitute of a worker, who performs his tasks in a particular environment. In this case, the model should be able to characterize the essence of "employment" in the sufficient way. One of the possible approaches to this problem is to treat the robot as an object, in the sense often used in computer science. This allows both: to describe certain operations performed on the object, as well as describing the operations performed by the object. This paper focuses mainly on the definition of the object as the model of the robot. This model is confronted with the other possible descriptions. The results can be further used during designing of the complete manufacturing system, which takes into account all the involved machines and has the form of an object-oriented model.

1. Introduction

Over the years, the role of industrial robots has changed in a meaningful way. Originally intended solely for manipulation, robots began to more actively participate in the production process, taking over the activities associated with work in hazardous environments (high temperature, fumes, etc.). This was mainly such works as painting of large surfaces (e.g., automobile sheets), welding and soldering, sandblasting etc. Nowadays the robots also carry out minor metal works like deburring, drilling, grinding or milling. Modern control systems allow using advanced vision systems, scanners, and tool changers. These entire features cause the robots increasingly replace a man at his workplace.

For this reason, their model must be considered in a similar way as a "worker" on a production line – more universal than any other machine and performing with different tools.

One of the possible methods of description of the robot's model is to treat the robot as an object in the virtual space. The paper discusses the main assumptions of the mentioned method as well as compares it to the others, existing methods of modelling.

2. Basics of the method

At the beginning of the 21st century, the methods based on agent systems began to grow in popularity [1-3]. Although the concept of an agent was already known, it had not significant practical application. It was the dynamic development of information technology that has caused a significant growth of interest in the subject among researchers. Nowadays agent systems are common, especially in IT solutions [3]. They are also used to model phenomena in human or animals populations that may be further adapted to the needs of robotics – particularly for control of mobile robots group. In this context, the return to the concept of the object as a model of the robot seems a bit archaic, but it is a deliberate action, because agents are – more or less – the natural development of the object model.
One of the most important features that characterize an agent is its fit in a particular environment [4, 5]. Referring this property to the machines, we could say that the agent intended for cold environment cannot act in the hot one; the agent intended for terrestrial use cannot fly etc. The agent is also considered as an autonomous, so its interaction with the surrounding world depends on its internal "intelligence". As a contrast, the object is more universal and general description. It is often depended on the other objects, so it cannot be considered as autonomous. The communication between object and the environment is rather determined, but sometimes the object is a "dumb" formation, which does not interact with its surroundings. The more differences that distinguish objects from agents are shown in the table 1.

Table 1. Some differences between objects and agents.

Property	Agent	Object
Environment	specific	undefined
Level of abstraction	higher (strict description)	high (general description)
Inheritance	not precluded from some form of inheritance	commonly used
Autonomy	autonomous	related
Type/class notion	none	commonly used

The distinction between the agent and the object becomes clear when we ask about the role, which the object should to fulfill. For example, if someone has a transport company, he also has trucks. The trucks could be of different types, like dump truck, platform, container, refrigerator truck etc. He also could have some busses to transport people. Speaking broadly, we see that this man has some vehicles. If we want to describe these vehicles as an object, we could define the class *vehicle* and create the several objects that are distinguished by purpose (figure 1): for example *container_truck*, *bus* or *minibus*. Every object will be a child of *vehicle* class and will inherit its properties and methods – the examples of properties could be a color, engine type or vehicle mass, while the example methods could be drive, engine start etc. Treating the vehicle as an agent is slightly different. Considering the definition of the agent, we should treat every vehicle as an autonomous subject. In that way, it is easy to see what the agent could "say" about itself, or better – what the driver of the particular vehicle could say. We could collect the information such the position of vehicle, if it is duty or free, the amount of fuel, failures etc. The information could be stored in the management center, which – in turn – will collate them with the customers' demands (figure 2). The lines correspond to the communication abilities of each agent – it could send messages not only to the center, but also to the other agent.

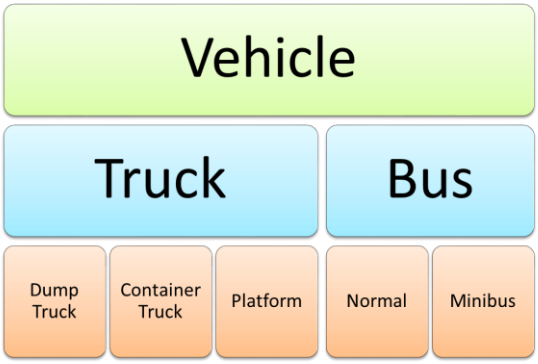

Figure 1. Simple illustration of object tree.

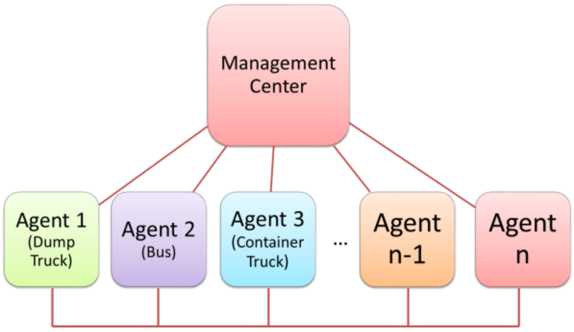

Figure 2. Simple organizational scheme of agent-based system.

As it can be seen, the basic distinction between the agent and the object depends primarily on approach [3, 6, 7]. It is therefore possible to treat an object as a base for creating the agent. This property will be used in further part of discussion.

3. The Object as a Model of a Robot

In the earlier discussion, the robot has been identified as the most complicated part of the production line. For this reason, it is difficult to find the right model that describes the robot in a universal way. Of course, the agent model could be taken into consideration: it contains all the information needed to specify the type of robot and control it. On the other hand this model applies only to the specific case and the concrete environment. In practice, the idea is to make the model of the robot as general as possible. The further upgrade of the model – making it more detailed – will be possible by adding more elements to an existing “skeleton”. This approach corresponds to the application of model in the form of object [8-11], along with the implications in the form of inheriting of properties and methods.

Assuming that the object is basically the set of properties and methods (equation (1)),

$$O = \{P, M\}, \tag{1}$$

where the P stands for properties and M for methods, we could create the class named *Robot* in a similar way (equation (2)), adding the *name* attribute that distinguishes the particular robot from others :

$$\begin{aligned} &Robot = \{P, M\}, \\ &name \in P. \end{aligned} \tag{2}$$

The properties and methods can be used within the meaning of features that characterize a particular model of the robot, but it is difficult to use them at a high level of generality. It must be recognized that, in principle, it is impossible to choose all the features of an industrial robot at the stage of planning the production process. In this case, we can rely on the scope of activities that the robot will perform and use them as the basis to define the subsequent requirements. Speaking in a very general way, the kinematic chain of the robot could be the representation of the properties, while the control system could represent the methods.

4. Building an Object-based Model of a Robot

The first stage of designing of the production line, when the particular task is assigned to the particular machine (equation (3)):

$$f : T \rightarrow Mch \tag{3}$$

where T is the set of tasks and Mch the set of machines. Assuming that some tasks will be realized by the robot or by the group of robots, we should assign the task to the robot (equation (4)):

$$\exists t \in (T_1 \subset T) \wedge \exists r \in (R \subset Mch) : \varphi(t, r) \tag{4}$$

where T_1 is a subset of tasks that are realized by robot, R is a set of robots and $\varphi(t, r)$ means that the task t is realized by robot r.

At this moment, the tasks are assigned to the robot (or robots), but the sets of properties and method of the *Robot* class are empty. In order to accomplish the task we should fill the set of properties P and set of method M in the manner that (equation (5))

$$\forall [(r \in R) \wedge (t \in T_R)] \exists [(p \in P) \wedge (m \in M)] \Rightarrow \varphi(t, r) \tag{5}$$

where $T_R \in T$ is a subset of tasks realized by the particular robot. The equation 5 is based on assumption that the robot could realize the tasks from the T_R subset (see equation (4)).

5. Relationship Between Real Robot and its Object-based Model

As a result of conduction of the operations mentioned in the previous section, the complete model of the robot is obtained in the form of the object, but depending on the assumed level of details, the resulting models will differ.

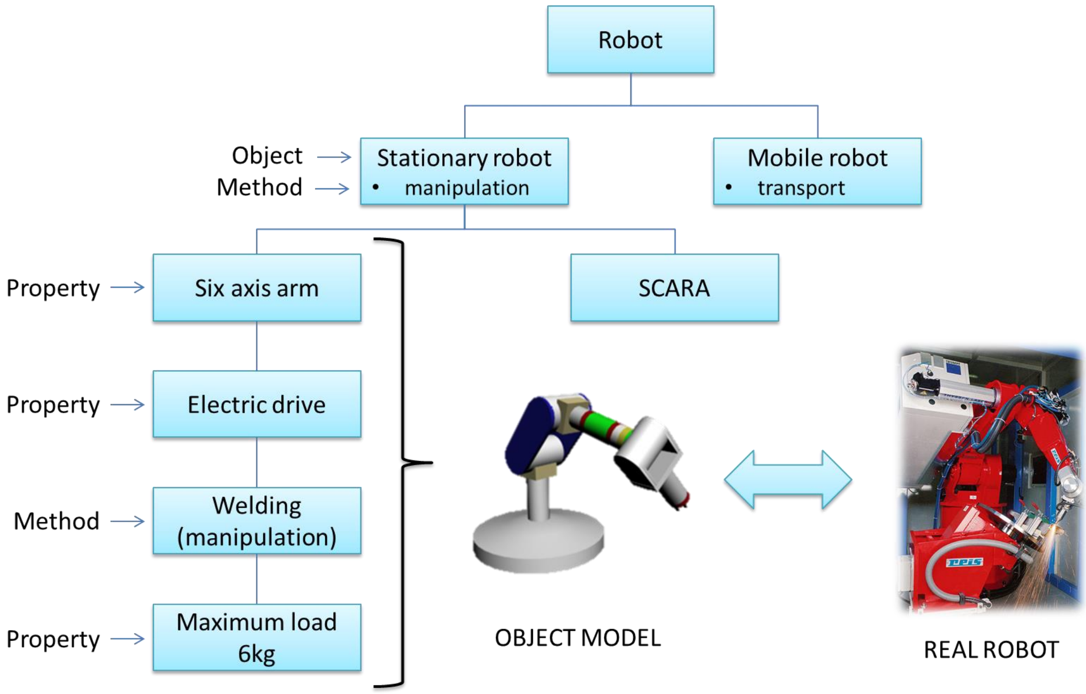

Figure 3. Simplified process of building the object-based model and its reference to real robot, [12-13].

Successive stages of assignment of another attributes leads to the complete characteristic of the robot. After completion of the process, the generalized model of the robot is obtained. In fact, due to its specific role and fit in a particular environment, this model can be seen as an agent. The final phase of the whole process is selection of a particular model of the robot from a specific manufacturer (figure 3).

6. Conclusions

Considering the production process we should pay attention to the things that are important on particular stage of designing the production line. First of all it is essential to define the types of machines that are involved in the whole process. Next the properties should be considered and eventually the way of communication (in the meaning of "language") between all of the devices. Because of its features, the robot should be seen as the very versatile machine that resembles the worker. For this reason, it is difficult to develop a model that will describe various application of the particular robot. In this paper the object-based approach was presented, as the one of possible ways of covering this problem.

The method shown in the paper, proposes the creation of abstract class named *Robot*, which could represent a real robot that acts in the production process. Further objects, which are created, are based on this class and equipped with the appropriate attributes and methods that allow fulfilling the specific tasks. The model, which is obtained in this manner, represents a hypothetical robot involved in the considered manufacturing process. Such a description is closer to the concept of an agent than an object, but the process leading to a final model is different – in the case of the object we are dealing with the evolution of the model and the creation of new forms based on previous results (inheritance).

The future use of the developed form of the robot's model could be connected with designing of manufacturing systems or robotized cells, using selected aspects of agent theory and object-oriented description.

Acknowledgements

The work is realized within the project titled: "Modular automated production stand with instrumentation for non-invasive confirmation of product quality" funded by The National Centre for Research and Development, agreement No UOD-DEM-1-495/001.

References

[1] Kubera Y, Mathieu P and Picault S 2010 Everything can be Agent! *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems* **1** pp 1547-1548

[2] Wood M F and DeLoach S A 2001 An overview of the multiagent systems engineering methodology *Agent-Oriented Software Engineering* **207-221** (Berlin: Springer Berlin Heidelberg)

[3] Jennings N R 2001 An agent-based approach for building complex software systems *Communications of the ACM* **44(4)** pp 35-41

[4] 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* (Berlin: Springer Berlin Heidelberg) pp 21-35

[5] Yan Z, Jouandeau N and Cherif A A 2013 A survey and analysis of multi-robot coordination *International Journal of Advanced Robotic Systems* **10**

[6] Sheremetov L B and Smirnov A V 1999 Component integration framework for manufacturing systems re-engineering: agent and object approach *Robotics and autonomous systems* **27(1)** pp 77-89

[7] Smimov A V and Sheremetov L B 1998 Agent & object-based manufacturing systems re-engineering: A case study *Changing the Ways We Work* pp 369-378

[8] Booch G 1986 Object-oriented development. Software Engineering *IEEE Transactions* **2** pp 211-221

[9] Rumbaugh J Blaha M Premerlani W Eddy F Lorensen W E 1991 Object-oriented modeling and design **199(1)** (Englewood Cliffs: Prentice-hall)
[10] Tummescheit H 2002 Design and implementation of object-oriented model libraries using modelica *Doctoral dissertation, Lund University*
[11] Otter M Elmqvist H Cellier F E 1996 Modeling of multibody systems with the object-oriented modeling language Dymola *Nonlinear Dynamics* **9(1-2**) pp 91-112
[12] RRU-Marketing, Laser Robot, CC-BY 3.0 Licence 2015 Retrieved from http://commons.wikimedia.org/wiki/File:Reis_Laser.jpg
[13] NeD80, 2015 3D Model of robotic arm, Public Domain, Retrieved from http://commons.wikimedia.org/wiki/File:Robot_arm_model_1.png