

# A Fuzzy Multiagent Approach for Integrated Product Life Cycle Environment

V.V. Taratukhin, Y.V. Yadgarova and E.Y. Skachko

**Abstract** One of the main questions in product life cycle management is how to create the comprehensive framework for autonomous, intelligent decision-making which integrates business and scheduling data. The key problem is to simulate human-like decision-making process to provide agile manufacturing process. Multiagent technologies play a key role in this problem and form an integration platform between human and manufacturing. Also developing distributed control system based on multiagent technologies encounters difficulties, due to ambiguous, vague, or missing information. In the area of intelligent manufacturing systems, there are a number of fuzzy scheduling models presented (Subramaniam et al. in *Int J Adv Manuf Technol* 16(10): 759–764, [1]; Srinoi et al. in *Int J Prod Res* 44 (11): 1–21, [2]). These frameworks only deal with manufacturing processes. The research presents multiagent framework that integrates design, manufacturing, and control process in fuzzy area using resource-based approach to agent's interaction. This model is applicable to work with various manufacturing agents in conjunction with different design agents and control systems.

---

V.V. Taratukhin (✉)

ERCIS Competence Center, University of Münster, Leonardo-Campus 3,  
48149 Münster, Germany  
e-mail: victor.taratoukhine@ercis.uni-muenster.de

Y.V. Yadgarova · E.Y. Skachko

Department of Computer Systems and Complex Automation, Bauman Moscow  
State Technical University, 2-ya Baumanskaya, 5, Moscow, Russia  
e-mail: y.v.yadgarova@gmail.com

E.Y. Skachko

e-mail: ekaterina\_skachko@inbox.ru

## 1 Introduction

On the one hand, developing a distributed control system that can operate with ambiguous and vague information is still an open problem, due to its complexity. But with rapid progress of computer capacity, new earlier unavailable technologies and solutions are taken into account.

On the other hand, the high-value products and infrastructure require engineering services, such as maintenance support throughout the life cycle. The future product life cycle management system should provide loosely coupled integration with design process (productive maintenance support on the stage of design, control of the design mistakes, and feedback) [3].

Smart industry, a key trend in manufacturing, is the ability of machines and devices to be self-organized, to communicate independently with each other, and to provide agile and adaptive design and manufacturing environment. One of the main ideas proposed in this paper is the new approach of communication and integration of the engineering design, manufacturing and planning systems' fuzzy self-oriented entities.

This paper proposes a new way of organizing manufacturing/control and design scene by including fuzzy logic approach in design and control subsystems. Unlike the traditional fuzzy multiagent scheduling systems, such system includes integration with design process agents to solve design process problems and fuzzy resource-goal approach to interact with agents.

## 2 Agent-Based Reconfigurable Model

The design principles of large-scale systems composed of heterogeneous working in real-time components require unified theory of real-time operations that includes existing results and novel solutions. Also, the cross-layer design is reached, specifying that each device should be designed based on hardware, operating system, middleware, sensing, actuation, as well as communication as a whole. One of the most appropriate technologies for creation such system is multiagent systems. The main benefits of multiagent systems are their decentralization and simplicity of development of agents.

In this paper, general three-layer architecture of multiagent system with design (D-), manufacturing (M-), and control (C-) agents including fuzzy logic approach of M-agent is presented. Resource-goal network approach is used for interaction of agents on M-layer, controlling and distributing resources among the agents.

Concept framework can be presented formally as follows:

$$CF = MP_1, \dots, MP_t, \dots, MP_n,$$

$MP_t$  is the  $t$ -manufacturing part,  $t = 1, 2, \dots, n$ .

$AP = DA_1, \dots, DA_i, \dots, DA_m, \dots, MA_1, \dots, MA_f, \dots, MA_k, \dots, CA_1, \dots, CA_j, \dots, CA_n$ .

## 2.1 Agent Types

For defining this framework, three types of agents were defined:

$DA_i$  is the  $i$ th Design Agent (D-agent),  $i = 1, 2, \dots, m$ .

$MA_f$  is the  $f$ th Manufacturing Agent (M-agent),  $f = 1, 2, \dots, k$ .

$CA_j$  is the  $j$ th Control Agent (C-agent),  $j = 1, 2, \dots, n$ .

Each  $DA_i$  consists of five elements: FB—facts base, which includes information about geometric characteristics of the part and material type; KB—knowledge base;  $K$ —corrector block, which adapts knowledge base, as a result of communications with any other agents;  $I$ —inference engine; and LI—local interface mechanism.

Manufacturing agent  $MA_f$  represents execution unit of the manufacturing system and consists of actuator and fuzzy multiagent control system. Fuzzy inference engine in each  $MA_f$  is responsible for routing and scheduling mechanism. Communication with each other implements with resource–goal fuzzy network approach.

Each  $CA_j$  consists of the following: MB—metaknowledge base; knowledge base of control agent; inference engine; corrector block; and local and global interface mechanism (GI).

On this layer, agents are using FIPA-compliant communication languages (ACL); global and local interface translates messages from external ACL to internal description and from internal description to external representation.  $K$ —corrector is realized for internal adaptation of knowledge and fact bases that will be described later in this report.

Each M-agent operates as an independent entity and processes associated manufacturing parts.

## 2.2 Resource–Goal Network Approach

Resource–goal network [4] as formalism for the MAS specification integrated to manufacturing environment is weighted directed graph:

$$NET_{RG} = (A, C, K, RES, G, W, t)$$

where set of vertex is a set of agents  $A$ , set of arcs  $C$  consist of goal links  $C_G$  and resource links  $C_{RES}$ ,  $C = C_G \cup C_{RES}$ ,  $C_G \cap C_{RES} = \emptyset$ ,  $K$ —types of agents  $A$ ,  $RES$ —set of resources in MAS,  $G$ —set of goals,  $W$ —set of conduction for resources and goals, and  $t$ —set of discrete time moments.

Every agent as a vertex of resource–goal network  $a_i \in A$  characterized by type of agent  $k_j \in K$  and amount of resources  $RES(a_i) \in RES$ . Every vertex  $c_{ij} \in C$  is characterized by conductivity (capacity)  $w_{ij} \in W$ . Conductivity of agent defines the maximum value of resource that one agent can transfer to another during one time interval. For resource–goal network is defined weight matrix along arcs  $W_c = [w_c(a_i a_j)]$  and weight matrix along resources  $W_{res} = [w_{res}(a_i a_j)]$ .

Decision-making of the agent about the use of resources or relocating it in this network is based on negotiations between agent and network characteristics at the moment.

### 3 Design Agent: Inference

To include fuzzy logic approach in MA-set fuzzy resource–goal routing among the manufacturing agents (MA) within order processing restrictions is implemented. Reconfigurable model works with several similar machines and tools that control the agent are able to vary in order to achieve the optimal route.

On the other hand, another restriction of the model is mismatching that can occur within manufacturing process. Design is multidisciplinary process and involves several stages such as conceptual design, basic structural design, detail design, production design, manufacturing processes analysis, and documentation. Design agent (DA) must detect and resolve two main types of mismatches:

- mismatches of integration
- concurrency mismatches [5–11].

Mismatches of integration are assembly mismatches. These types of mismatches detect geometric, material characteristics of the part for checking assembly possibility.

According to [12], “Concurrent engineering is getting the right people together at the right time to identify and resolve design problems. Concurrent engineering is designing for assembly, availability, cost, customer satisfaction, maintainability, manageability, manufacturability, operability, performance, quality, risk, safety, schedule, social acceptability, and all other attributes of the product.”

Concurrency mismatches are controlled by the design agents. Such type of agent stores knowledge base, facts base, and inference engine. There is information about sequences part’s structure parameters. Details of D-agents are presented in [13].

## 4 Manufacturing Agent: Fuzzy Resource–Goal Network Approach

Configuration of the manufacturing system consists of warehouse ( $W$ ), robocara ( $R$ ), two milling machines ( $MM_1, MM_2$ ), two turning lathe ( $TL_1, TL_2$ ), two industrial robots ( $M_1, M_2$ ) and control system ( $C$ ). Parts require a different set of operations  $\{op_1, op_2, op_3, \dots\}$  and can be routed for several different ways according to processing route and weight coefficient of link between agents.

Resource–goal network as a model of agent’s interaction is based on matrix of goals and matrix of resources. To interact with other agents and route parts in different stages of manufacturing process, agents must exchange information about their resources, goals, and state of the environment. However, the problems occur when information about agent’s resources or state of the environment is incomplete, ambiguous, and vague. One way of solving this problem is using fuzzy logic approach to present resource’s state of agent. Unlike the classical logic propositions in fuzzy logic are not limited to one of the two values True, False<sub>1</sub> (or 0, 1) [14].

Every two agents  $a_i$  and  $a_j$  in resource–goal network has **Goal conductivity**  $w_c(a_i a_j)$  and **Resource conductivity**  $w_{res}(a_i a_j)$ .

To include fuzzy logic approach in resource–goal formalism is defined fuzzy variables [15]:

**Fuzzy resource conductivity** indicates capacity of the vertex in the goal–resource network. Physical sense of this variable is the maximum value of resource that one agent can transfer to another during one time interval.

$$FRC_{ij} = \text{minimum, low, middle, large, max}$$

If  $FRC_{ij}$  of two agents is known, it is possible to define **Weight matrix** along resources in manufacturing system.

For example, fuzzy weight matrix between three actuators can be defined as:

$$\begin{pmatrix} \text{low} & \text{middle} & \text{low} \\ \text{middle} & \text{high} & \text{high} \\ \text{low} & \text{middle} & \text{low} \end{pmatrix}$$

### Fuzzy resource amount

$$FRC = \text{low, middle, large}$$

### Fuzzy workload

$$FW = \text{low, middle, large}$$

**Fuzzy workload** of the  $MA_f$  indicates quantity of parts requested for this agent. Quantitative representation of this value is queue size of the  $MA_f$ .

Generally, fuzzy sets depend on the type of the manufacturing: Resource conductivity of industrial robot and milling tool is different. Also workload level of each actuator is different. In this work, it is considered that parts group by similar resource requirements.

As the fuzzy output variable is defined **Fuzzy Requirement** variable represented by sets:

$$FR_{ij} = \text{low, average, high, critical}$$

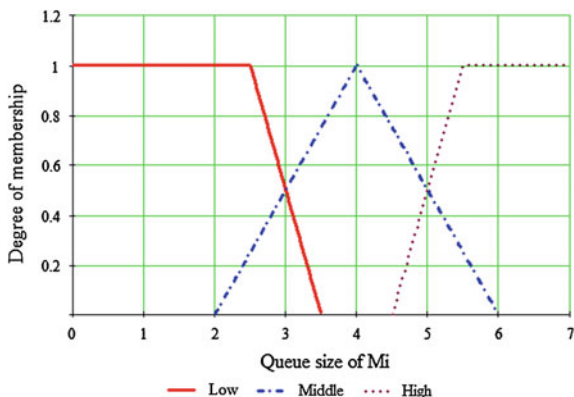
**Fuzzy Requirement** variable shows demand of resource that  $MA_i$  should transfer to  $MA_j$ .

For every pair of manufacturing agents  $MA_f$  and  $MA_j$ , conditional fuzzy propositions were presented in this form :

$$\text{If } FRC_{ij} \text{ is } * \text{ and } FW_i \text{ is } * \text{ and } FRA_j \text{ is } * \text{ then } FR_{ij} \text{ is } *$$

The set of fuzzy rules consists of 27 ( $3 * 3 * 3$ ) rules. This set is constructed for every  $FR_{ij}$  variable. The simple subset for two manufacturing agents is presented on Fig. 1 (Tables 1 and 2).

**Fig. 1** Fuzzy sets for  $FW_i$  variable



**Table 1** Fuzzy variables

Linguistic variable	Fuzzy set
$FRC_{ij}$	Minimum, low, middle, large, maximum
$FW_i$	Low, middle, large
$FRA_i$	Low, middle, large
$FR_{ij}$	Low, average, high, critical

**Table 2** Fuzzy rules for  $FR_{ij}$  variable

$FR_{ij}$	$FW_i$	$FRA_j$	$FRC_{ij}$
Average	Low	High	Middle
Average	Low	Middle	Large
Critical	High	High	Minimum
High	High	Middle	Middle
Low	High	Low	Middle

Resource demands information required to solve problems:

- Routing parts to actuators with satisfied resource demands and high value of workload;
- Normalize amount of resources for  $MA_f$ .

Final routing between actuators provided by  $CA_i$  combined fuzzy inference engine and depends on manufacturing process, collisions during processing, and resource demands of the  $MA_f$ .

## 5 Control Agent: Inference

Control agent stores metaknowledge values about technological environment such as map of the production system, distance between machines, stages of the design process, and overall design time. Also control agent stores actual knowledge base (amount, state and location of the parts, and resource demands of  $MA_f$ ) and overall state of environment.

One of the main functions of the control agent is final routing parts through manufacturing line. Based on manufacturing agent’s workload, resource demands, state and design agent’s information, a combined inference engine of this agent can be defined (Fig. 2).

Knowledge base in control agent is responsible for order processing and routing parts between manufacturing agents. Generally, order processing in this case is strictly defined by technologist, but in case of several same machine tools, control agent must route part on the least loaded machine tool with low demand of resources. So, order processing is the restriction of our model.

According to mismatches restrictions in design agent, intermediate route has been produced. Then on the final step, selected routings are the input to the control agent that produces string route of the part based on its metaknowledge base and state of the part.

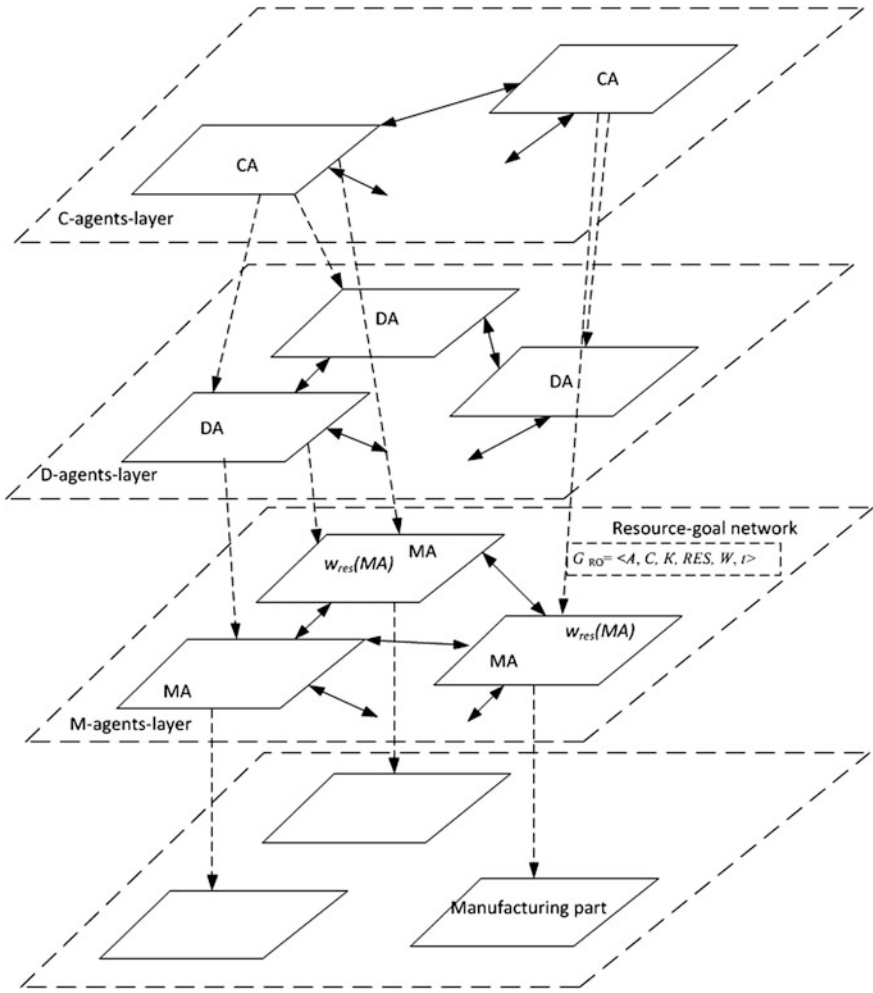


Fig. 2 Multiagent framework architecture

## 6 Prototype Implementation

In this research, a prototype of the control system which is capable of integrating with hardware devices is proposed. Such subsystem provides flexible configuration of each device. Rules of communication, ontology, knowledge base, and facts base are configured separately by an engineer of corresponding domain.

The initial multiagent system consists of different types of agents, group of agents, and objects. In the designed subsystem, the universal circuit board can be embedded in any device, controlled by a UNIX class operating system which contains an agent platform.



As a manufacturing agent, an implementation with RFID readers to store real-time information about the product is used. Such information (destination, state, and location of the part) influences the control agent inference engine.

### 6.1 Hardware Overview

Prototype of the control system consists of a single board computer, RFID reader, and actuator. The scheme of the control system is presented on the Fig. 3.

The single board computer holds UNIX OS on board, provides the software for controlling the agent, and power supply to the whole system.

The RFID reader is connected by means of SPI (serial peripheral interface) via development board. This interface proves to be the most suitable for the kind of task, which the constructed device (the agent) fulfills. The interface can handle data speeds up to 10 Mbit/s. It works in full-duplex mode.

It also allows arbitrary choice of message size, content, and purpose. SPI is preferable for energy consumption factor as well, because in the current design, the device's peripherals do not use any outer power supply. Devices communicate in master/slave mode, where the master device initiates the data frame. In the present design, the microcontroller of the single board computer plays a role of master device.

The exemplary executive device is Mitsubishi industrial robot (RV-1A series). It is a compact 6-axis robot-manipulator, which gives excellent options for building a flexible manufacturing system. You can see the device in Fig. 4. The robots controller supports RS-232 interface, so TxD and RxD pins of the single board computer are used to connect it.

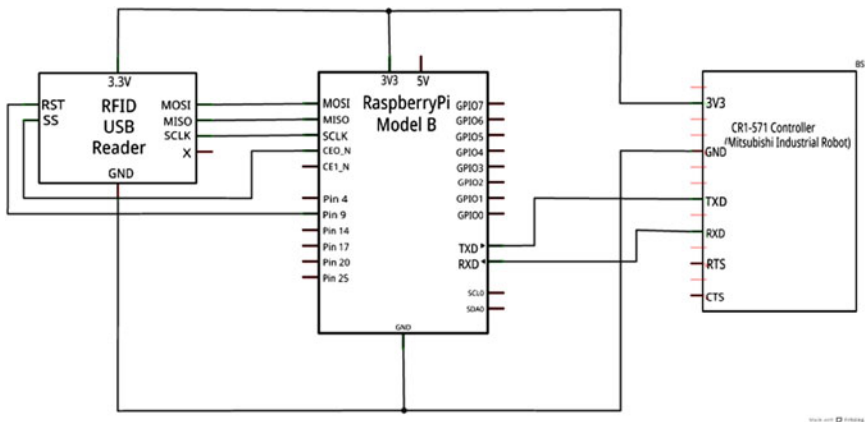


Fig. 3 Schema of the control/routing device



Fig. 4 Mitsubishi industrial robot

## 6.2 *Software Overview*

After hardware connection, there is a need to set a convenient software interface of communicating with RFID. The reader requires interface with registers and its native instructions. From the point of high-level code, it is inconvenient. Another level of abstraction for communicating with RFID is needed.

In order to control and to interact with RFID reader, a third-party Python library is used. Python allows importing GPIO and SPI wrapper. It hides low-level abstractions and provides quite a convenient interface of communication with RFID.

It operates 64 registers of the RFID device and uses native instructions, common for SPI interaction.

The functions of software are as follows:

- Reading part's data and store current information of it's state;
- Updating data during processing;
- Realization of resource–goal formalism;
- Communicating with other  $MA_f$  and  $CA_j$ .

## 7 Conclusions

In this paper, a new multiagent framework with fuzzy logic approach was developed. This framework includes distributed control system based on multiagent technologies which allows solving problems related with ambiguous or missing information. The research presents approach that integrates design, manufacturing, and control process using resource-based fuzzy network to agent's interaction. Prototype of the system is presented. This model is applicable to work with various manufacturing agents in conjunction with different design agents and control systems.

## References

1. Subramaniam, V. et al.: Job shop scheduling with dynamic fuzzy selection of dispatching rules. *Int. J. Adv. Manuf. Technol.* **16**(10), 759–764 (2000)
2. Srinoi, P., Shayan, E., Ghotb, F.: Scheduling of flexible manufacturing systems using fuzzy logic. *Int. J. Prod. Res.* **44**(11), 1–21 (2002)
3. Taratoukhine, V., Bechkoum, K.: Towards a consistent distributed design: a multi-agent approach. In: *Proceedings. IEEE International Conference on Information Visualization*, pp. 384–389 (1999)
4. Dyundyukov, V., Tarassov, V. Goal-resource networks and their application to agents communication and co-ordination in virtual enterprises. *Manuf. Model. Manage. Control* **7**(1), 347–352 (2013)
5. Kandel, A.: *Fuzzy Mathematical Techniques with Applications* (1986)
6. D'Ambrosio, J., Darr, T., Birmingham, W.: Hierarchical concurrent engineering in a multiagent framework. *Concurrent Eng.* **4**(1), 47–57 (1996)
7. Szczerbicki, E.: Distributed multiagent system for concurrency evaluation: conceptual architecture. *Syst. Anal. Model. Simulation* **16**(4), 277–291 (1994)
8. Pham, D.T., Dimov, S.S.: An approach to concurrent engineering. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **212**(1), 13–27 (1998)
9. Tong, G., Fitzgerald, B.: Concurrent engineering battleground. *World Class Design Manuf.* **1**(3), 45–48 (1994)
10. Danesh, M. R., Jin, Y.: An agent-based decision network for concurrent engineering design. *Concurrent Eng.* **9**(1), 37–47 (2001)
11. Sun, J., Zhang, Y.F., Nee, A.Y.C.: Agent-based product design and planning for distributed concurrent engineering. In: *Proceedings of Robotics and Automation. ICRA'00. IEEE International Conference on, IEEE*, vol. 4, pp. 3101–3106, 2000

12. Dean, E.B., Unal, R.: Elements of designing for cost. In: Proceedings of AIAA 1992 Aerospace Design Conference, 1992
13. Tarchinskaya, E., Taratoukhine, V., Matzner, M.: Cloud-based engineering design and manufacturing: state of the art. *Manuf. Model. Manage. Control* **7**(1), 335–340 (2013)
14. Prasad, B.: *Concurrent Engineering Fundamentals*, vol. 1. Prentice Hall, Englewood Cliffs, NJ (1996)
15. Zadeh, L. A.: The Concept of a Linguistic Variable and Its Application to Approximate Reasoning, pp. 1–10. Springer, US (1974)