





# Multi-agent-Based Approach for Complex Industrial Process Modeling

Kenza Redjimi<sup>(✉)</sup>  and Mohammed Redjimi 

Faculty of Science, Department of Computer Science, LICUS Laboratory,  
University 20 August 1955, 21000 Skikda, Algeria  
{k.redjimi,m.redjimi}@univ-skikda.dz

**Abstract.** A large number of components that interact with each other characterize complex systems. The modeling of such systems is generally based on a fine division of sub-assemblies of components of less complexity. Each component can be individually modeled and then simulated. This article deals with a top-down modeling method using multi-agent systems. The overall system is considered as a set of interacting agents. The behavior of the global system emerges from the behavior of its parts. This approach was applied to the liquefied natural gas treatment process (decarbonation) at an Algeria gas plant. The results of the simulation can be used to train recruits in this factory.

**Keywords:** Multi-Agent System (MAS) · Industrial system modeling and simulation · Agent/Group/Role (AGR) Model · Aalaadin · Madkit

## 1 Introduction

Today, industrial systems employ advanced technologies for the development of complex industrial processes. The vast majority of these systems can be considered complex in the sense that they are composed of a large number of interacting elements. To develop such systems, to predict their behaviors before beginning to build them, or for educational training, these systems are modeled and undergo a set of tests.

The use of computer tools for modeling offers the advantage of building a computer model that is very faithful to the considered system, a set of its parts, then to judge the behavior of the obtained model by injecting it with a set of data as inputs. This makes it possible to handle, observe, and improve understanding of its behavior and allows avoiding any possible errors concerning its future evolutions. The shutdown of certain large so-called critical systems can cause material and financial damages. For these systems, the use of models representing them is of substantial help. In fact, instead of stopping the operation of all or part of a plant, it suffices to consider the models, which represent them. This is the case, for example, with industrial boilers, the shutdown of which can cause the shutdown of a whole part of the plant, which depends on it.

The notion of system is at the center of several fields and disciplines. A system is characterized by a set of communicating and interacting elements. A system receives inputs and delivers results. Various systems can be envisaged, such as industrial systems,

economic systems, political systems, etc. A metric often used to characterize a system concerns its complexity.

The modeling of systems has been the subject of many studies and researches considering its important impact on the phenomena implemented. P. Fishwick's [1] considers modeling as a set of metaphors and analogies to better understand any phenomena. For C. Oussalah's [2], modeling is a transformation between a system and a model to give a simplified and observable representation of its structure and behavior.

The experimentation of the model can be considered as its simulation. Practically, a simulation is the execution phase of the model [3]. Given a set of pertinent data as input of the model, this phase consists of evolving the model and observing and analyzing the obtained results. So, works can be observed as modeling of actual cases in the laboratory.

Simulation of complex systems can take hours or even days and consume a lot of CPU time. This is why modelers have recourse to methods of reduction of complexity. Among these methods, there are parallel simulation [4], decomposition of the phenomenon in several entities of fewer complexities [5], and the use of fast and parallel CPUs [6].

The search for high fidelity, precision, and adaptation requires modeling and simulation data processing tools to be chosen with great care. Their use in the field of modeling and simulation is very widespread, ensuring excellent qualities for which the improvements are continuous with reasonable costs.

Several modeling approaches, methods, and tools have been and continue to be used in the literature. Petri nets (PN) [7] are among the most efficient and widely used tools. Despite their 62 years of existence, they continue to be relevant today.

These networks, despite their simplicity, make it possible to represent the dynamics of systems. Several variations and extensions of these networks have been developed, such as Colored PN (CPN) and Timed PN [8, 9].

The UML statechart diagrams are other powerful semi-formal tools that can represent the behavior of systems [10]. At a higher level, multi-agent systems (MAS) are modeling tools that offer the advantage of a fine decomposition of a real complex system into a set of its components. The behavior of the global system emerges from that of the behaviors of its entities. The notion of agent designates an entity whose functioning is autonomous and which is capable of carrying out a set of tasks [11, 12]. These systems are used more and more in computer systems because of their flexibility, simplicity, effectiveness, and possibility of extension. Moreover, a human observer can easily understand the description given by MAS.

The rest of this paper is organized as follows: Sect. 2 presents the multi-agent systems-based approach. Section 3 is dedicated to the proposed approach. Section 4 introduces a use case of industrial system simulation, and finally, Sect. 5 concludes this paper.

## 2 The Multi-agent System (MAS)

Multi-agent systems (MAS) provide high-level tools for modeling distributed systems. Nowadays, their application has become very widespread, and they have been applied with success in various domains, especially those based on artificial intelligence. A set of autonomous agents cooperate and collaborate in the execution of a common objective. From the behavior of all these agents emerges the overall behavior of the system [11–20].

### 2.1 Agent Definition

There is no common agreement about a definition of the term agent. In general, an agent is an entity (hardware or software), which is different from an ordinary program in the sense that it is endowed with autonomy for functioning in environments shared by other agents [12, 16]. An agent is able to evolve in an environment through the acquisition of information emitted by the latter and by acting on it. An agent’s architecture varies from a simple structure formed of production rules often linked to very complex structures where the agent can be endowed with advanced cognition and reasoning mechanisms, referring to it as an associative and social character [21].

## 3 The Proposed Approach

The work concerned with this present theme is related to the development of industrial process simulators in the hydrocarbon sector. This involves providing educational support to technical staff so that they can understand the different stages of the process.

The proposed approach is a top-down approach, which starts, therefore, from the overall process and then divides it up according to the fundamental units (see Fig. 1) after having identified them.

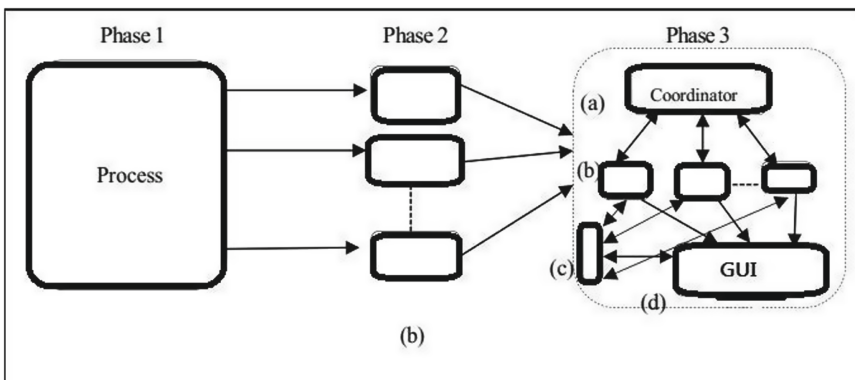


Fig. 1. Process breakdown.

The process cutting procedure is carried out as follows:

1. The process is studied in detail. The principle of its operation is clearly established.
2. The different units making up the process are identified according to the detailed technical diagram of the process (b).
3. Each unit is isolated to be individually modeled, and the obtained units are assembled in a global diagram.

Three additional control and communication units are added to the model:

1. A coordinator whose role is to control and manage the execution of the whole system.
2. An input interface through which the user can act on the system.
3. An interface for visualizing the progress of the process according to the inputs sent by the user.

### 3.1 Tools for the System Design

For the design of this kind of simulators, a solution using multi-agent systems is very appropriate. An agent represents each unit. The same goes for the coordinator, the input interface, as well as the display interface.

Nowadays, there are several multi-agent systems-based approaches well suited to the modeling and simulation of systems. Likewise, a wide variety of multi-agent platforms is available. Among the existing methodologies, the AGR (Agent-Group-Role) model gives very good results in terms of modularization of systems.

In the AGR model, an agent can play one or more roles in one or more groups. A group is made up of at least one agent. An agent who does not belong to a group can request to join this group. Several agents can play the same role. An agent can communicate with another only if the two agents are in the same group. The interactions between agents are founded on the messages exchange principle and allow coordination, collaboration, and synchronization between agents.

The AALAADIN meta-model is an organizational model for the modeling of multi-agent systems based on AGR concepts. AALAADIN provides a set of conceptual tools for the implementation of applications based on multi-agent systems. AALAADIN was developed at the computer science, robotics, and micro-electronics laboratory, Montpellier (LIRMM) [22–24].

### 3.2 The MADKIT Platform

The Multi-Agent Development Kit platform (MadKit) is a free software for the design and the simulation of Multi-Agent Systems (MAS) [25–29]. MadKit is a set of packages of Java classes. It implements a reduced size agent kernel, various libraries of messages, probes, and agents. In addition, Madkit includes a useful graphical development environment and many other agent handling systems. Madkit implements different tools for the management of agents, groups, and roles according to the AGR model. This platform is based on the AALAADIN meta-model.

## 4 Use Case

Pure natural gas (NG) is a fuel unfit for consumption due to the presence of impurities, particularly water and carbon dioxide (H<sub>2</sub>O and CO<sub>2</sub>), which reduces its calorific value. Liquefaction of NG requires cooling to about -165 °C. Knowing that the freezing temperature of the water is at 0° C and that of the CO<sub>2</sub> at -20 °C, this will clog the pipes of the machines during the liquefaction process. For these reasons, NG must be treated (decarbonized and dehydrated) before the liquefaction operation. Decarbonation is the extraction of carbon dioxide CO<sub>2</sub> from natural gas, and dehydration consists of extracting water from it. There are several industrial ways to execute the decarbonation. [30] presents the process adopted in this work.

### 4.1 The Natural Gas Decarbonation Process

In this use case, there are two operations for the decarbonation process: i) the washing operation, which is intended to extract the CO<sub>2</sub> from the NG with an MEA (C<sub>2</sub>H<sub>5</sub>ONH<sub>2</sub>), and the regeneration phase, where the results from the washing operation enter a regeneration tower so that they release CO<sub>2</sub> and then restore the MEA solution. Figures 2 and 3 illustrate the dedicated equipment.

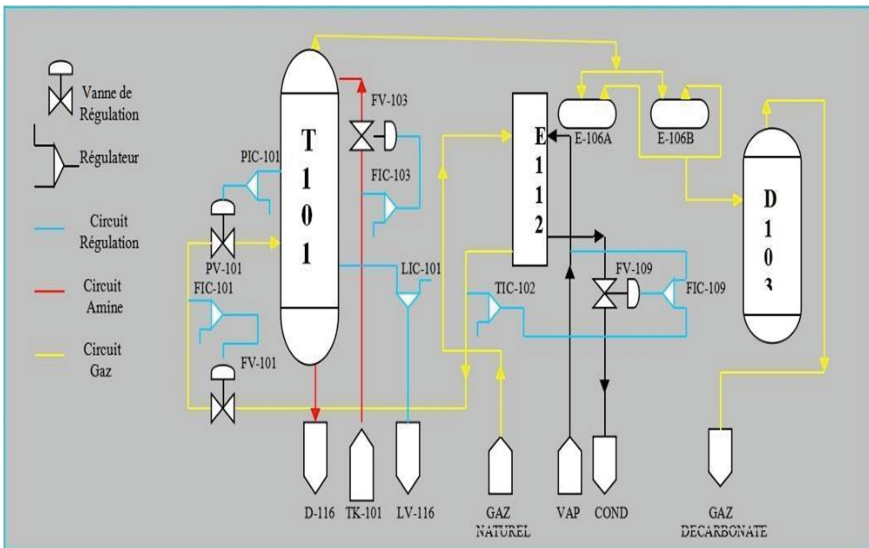
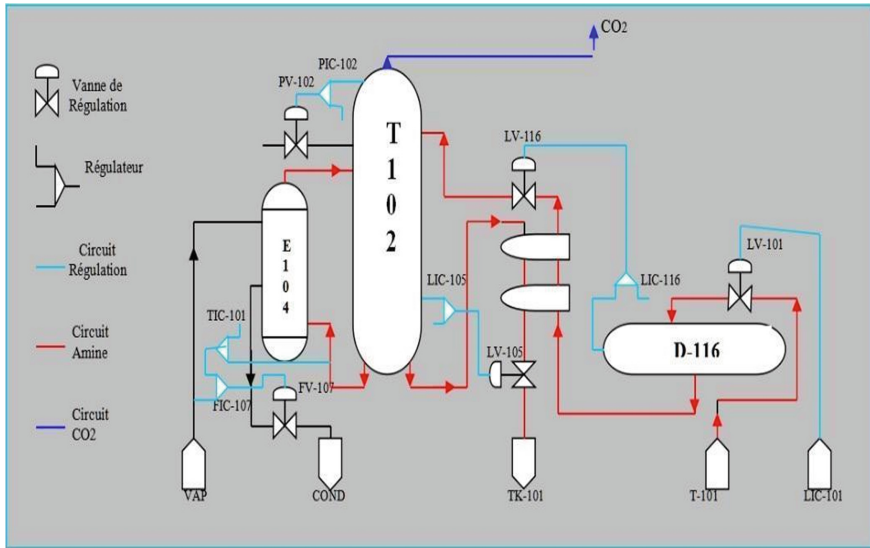


Fig. 2. Technical drawing of the washing process.



**Fig. 3.** Technical drawing of the regeneration process.

According to the breakdown system phases presented in Fig. 1, at first and after studying and analyzing the whole system as presented in Figs. 2 and 3, the following four points are highlighted:

- **The system ordered:** it is the decarbonation process. It contains a set of organs (T-101, T-102, D-116, E-112, E-104...) which carry out the various chemical reactions to achieve the decarbonation of the Natural Gas.
- **The control system:** this is the distributed control system for the natural gas decarbonation process. It is formed by a set of autonomous, modular, and distributed instruments. They are intended for the control and regulation of the process.

These instruments are divided into two sections:

- **The washing section:** it groups the regulation valves (FV-109, FV-103, FV-101, PV-101), which are regulated by a control system, which contains the different regulation loops.
- **The regeneration section:** also contains a set of control valves (FV-107, PV-102, LV-116, LV-105, LV-101), which are regulated by the same control system. To carry out this adjustment, the control system is based on the set of instructions.

## 4.2 The Proposed AGR Model

To build an interactive simulator for the above-mentioned industrial gas process, the three following groups, as depicted in Table 1, have been distinguished. Note that this point corresponds to phase 2 of the proposed simulation approach:

**Table 1.** Obtained agents, groups, and roles.

Id_Group	Id_Agent	Designation	Role
Perturbation	GUS	Graphical User System	To interact with the system
Washing	APM	Agent Process Manager	To acquire the Instructions
	APM	Agent Process Manager	To regulate the washing group
	FV-103	Flow Valve Agent	To control the Amine flow
	FV-101	Flow Valve Agent	To check the natural gas flow
	PV-101	Pressure Valve Agent	To check the pressure of T-101
	FV-109	Flow Valve Agent	To check the steam flow of E-102
Regeneration	APM	Agent Process Manager	To set the regeneration group
	FV-107	Flow Valve Agent	To check steam flow E-104
	PV-102	Pressure Valve Agent	To check the pressure T-102
	LV-101	Level Valve Agent	To check the level of T-101
	LV-116	Level Valve Agent	To check the level of D-116
	LV-105	Level Valve Agent	To check the level of T-102

This decomposition was obtained after a meticulous study of the NG decarbonation process as well as the material structure and the arrangement of the interactions between the different material components (pressure, level, flow control, etc.). This article will not go into the precise details of how the system works.

The APM agent ensures the time synchronization between the different units. This particular agent (APM) serves as a coordinator in the system.

## 4.3 The AALAADIN Model

The entire model with links between all the components (agents) of the system is depicted in Fig. 4 below. This diagram does not describe the roles of each agent. This step corresponds to phase 3 of the proposed demarche.

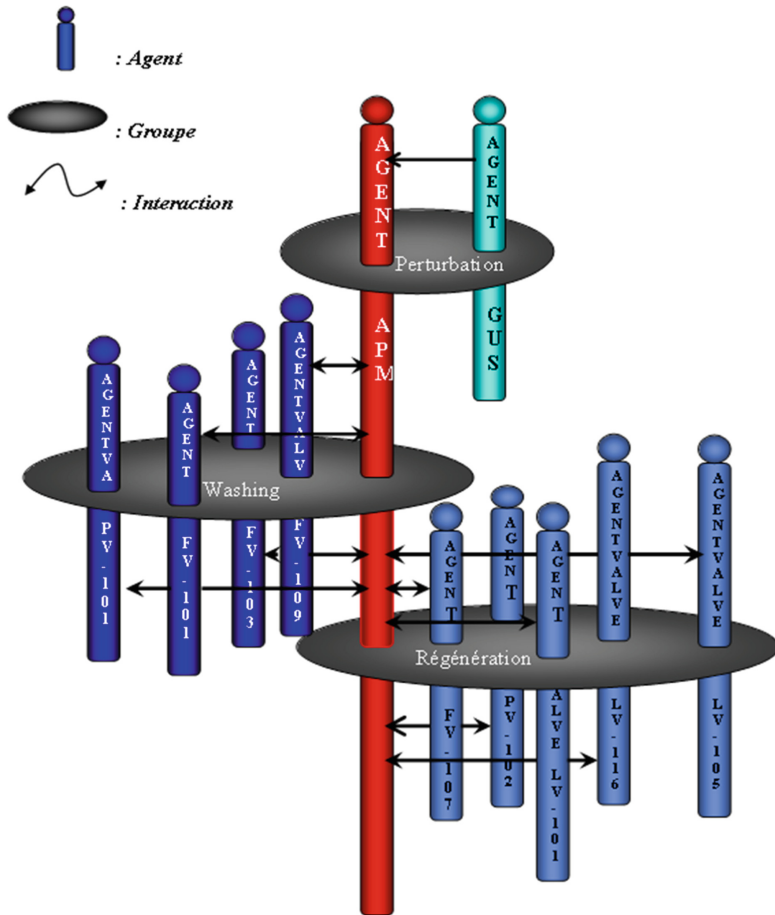


Fig. 4. The AALAADIN model of the simulator.

#### 4.4 The System Implementation

The simulator was developed by using the Mad-Kit agent platform based on AALAADIN model. The two main interfaces of the project are presented in Figs. 5 and 6. These two interfaces concern the washing section and the interface of the regeneration section. They allow the user to drive the washing and the regeneration operations and supply the system with data inputs.

#### 4.5 Discussion

This work aims to implement the MAS mechanisms of cooperation, collaboration, and synchronization between agents. These tools are used to design and to realize industrial simulators. The use case presented and implemented here, which can be very useful as pedagogical support, consolidates and validates the proposed approach.



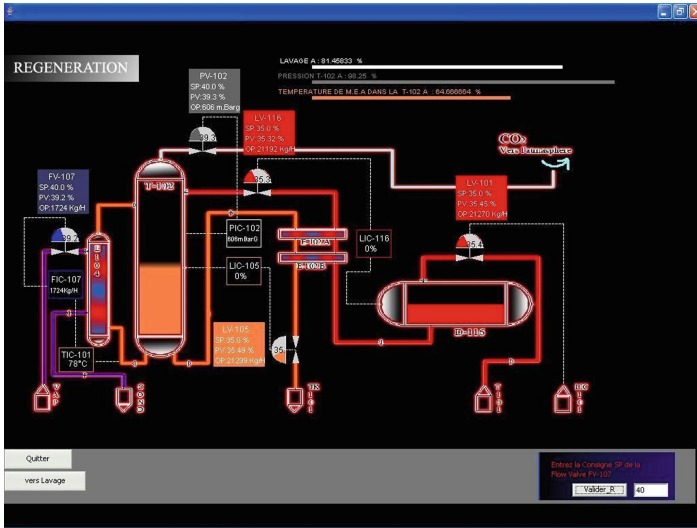


Fig. 5. The interface of the washing section simulator.

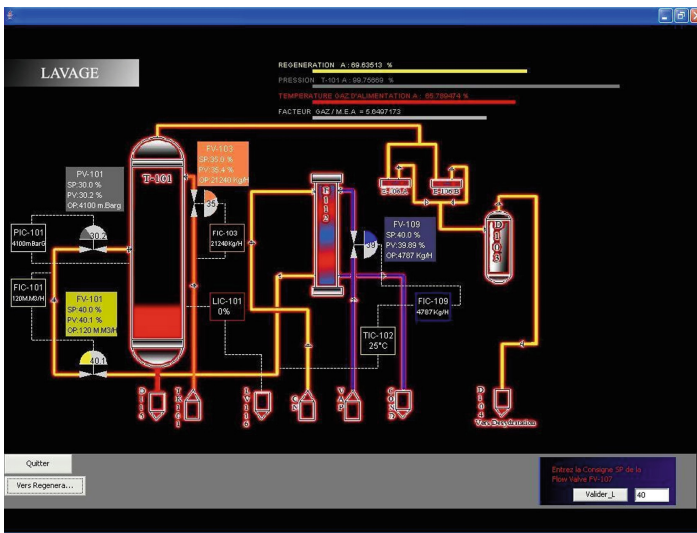


Fig. 6. The interface of the regeneration section simulator.

## 5 Conclusion

The concerns of this work are to introduce some notions of the design of industrial simulators. This work responds to a real need of providing a tool allowing young beginners in the company to understand the evolution of an industrial process. In this case, it is the process of liquefied natural gas decarbonation. The equipment allowing this procedure is modeled using agents. The coordination and cooperation between these agents allow

the exchange of information and the evolution of the process. This simulator is currently operational, and its use allows significant financial and material gains.

All the components of the system are modeled using the agent paradigm. The interaction between these agents ensures the evolving of the system, and the global behavior emerges from the local agent's execution. The use of consistent and coherent tools (for example, AALAADIN and MADKIT) is of great interest in obtaining efficient software products, as is the case here. In future work, we project to extend this approach to distributed systems modeling.

## References

1. Fishwick P (1995) Simulation model design and execution: building digital worlds
2. Oussalah C (1988) Modèles hiérarchisés multi-vues pour le support de raisonnement dans les domaines techniques. Technical report
3. Fishwick P, Paul A (1997) Computer simulation: growth through extension. *Trans Soc Comput Simul Int* 14(1):13–23
4. Mubarak M, Carothers CD, Ross RB, Carns P (2016) Enabling parallel simulation of large-scale HPC network systems. *IEEE Trans Parallel Distrib Syst* 28(1):87–100. <https://doi.org/10.1109/TPDS.2016.2543725>
5. Wainer G, Liu Q, Jafer S (2016) Parallel simulation of DEVS and Cell-DEVS models in PCD++. In: *Discrete-event modeling and simulation*, pp 223–270, 20115630
6. Martinez-del-Amor MA, Riscos-Núñez A, Pérez-Jiménez MJ (2017) A survey of parallel simulation of P systems with GPUs. *Bull Int. MC Soc* 55–68
7. Juranic J, Pavkovic N, Naumann T, Marjanovic D (2017) Modelling the design parameters dynamics with Petri nets. In: *DS 87-2 Proceedings of the 21st international conference on engineering design (ICED 17) Vol 2: design processes, design organisation and management, Vancouver, Canada*, pp 091–100
8. Jensen K, Kristensen LM (2009) Coloured petri nets. Springer, Heidelberg. <https://doi.org/10.1007/b95112>
9. Jensen K, Kristensen LM, Wells L (2007) Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int J Softw Tools Technol Transfer* 9(3–4):213–254. <https://doi.org/10.1007/s10009-007-0038-x>
10. Sunitha EV, Samuel P (2019) Automatic code generation from UML state chart diagrams. *IEEE Access* 7:8591–8608. <https://doi.org/10.1109/ACCESS.2018.2890791>
11. Mess M, Guerrits B (2018) Multi-agent Systems. *Lecture notes in logistics*, pp 611–636. <https://doi.org/10.1007/978-3-319-92447-2-27>
12. Ferber J (1995) Les systèmes multi-agents: vers une intelligence collective. Informatique, intelligence Artificielle, Intereditions Paris
13. Chen F, Ren W (2019) On the control of multi-agent systems: a survey. *Found Trends Syst Control* 6(4):339–499
14. Weiss G (1999) Multiagent systems – a modern approach to distribute artificial intelligence. MIT Press, Cambridge
15. Wang B, Wang J, Zhang B, Li X (2016) Global cooperative control framework for multiagent systems subject to actuator saturation with industrial applications. *IEEE Trans Syst Man Cybern Syst* 47(7):1270–1283
16. Ferber J (1999) Multi-agent systems. An introduction to distributed artificial intelligence. Addison-Wesley, Boston
17. Shoham Y (1993) Agent-oriented programming. *Artif Intell* 60(1):51–92

18. Franklin S, Gasser A (1997) Is it an agent, or just a program? A taxonomy for autonomous agents. In: Muller, Wooldridge, Jennings (eds) *Intelligent agents III. Agent theories, architectures, and languages*. Springer, pp 21–35
19. Wooldridge M, Jennings NR (1995) Agent theories, architectures, and languages. In: Wooldridge, Jennings (eds) *Intelligent agents*. Springer, pp 1–22
20. Russell S, Norvig P (2010) *Artificial Intelligence: A Modern Approach*. Pearson, London
21. Ye P, Wang S, Wang F (2018) A general cognitive architecture for agent-based modeling in artificial societies. *IEEE Trans Comput Soc Syst* 5(1):176–185. <https://doi.org/10.1109/TCSS.2017.2777602>
22. Ferber J, Gutknecht O, Michel F (2004) From agents to organizations, an organizational view of multi-agent systems. In: *Agent oriented software engineering (AOSE) IV*, vol 2935, pp 214–230
23. Soudi MEH, Songhao P, Guo L, Lin C (2016) Multi-agent cooperation pursuit based on an extension of AALAADIN organizational model. *J Exp Theoret Artif Intell* 28(6):1075–1088. <https://doi.org/10.1080/0952813X.2015.1056241>
24. Russell SJ (1997) Rationality and intelligence. *Artif Intell* 94:57–77
25. Ferber J, Gutknecht O (1998) Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems. In: Demazeau Y (ed) *ICMAS (International conference on multi-agent systems)*, Paris. IEEE Press, pp 128–135
26. Chebout MS, Mokhati F, Badri M, Babahenini MC (2019) Monitoring open multi-agent systems: an aspect-oriented programming based approach. *Multi-agent Grid Syst* 15(2):155–177. <https://doi.org/10.3233/MGS-190307>
27. <https://mansour.saber.free.fr/termadkit/site/madkit/doc/devguide/devguide.html>
28. Gutknecht O, Ferber J (2000) The MadKit agent platform architecture. Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier
29. <https://www.madkit.net/madkit>
30. <https://patents.google.com/patent/WO2014199036A2/en>