Multi-Agent Modeling of a Complex System

Noureddine Seddari, Mohammed Redjimi
Department of Computer Science
20 August 1955 University
Skikda, Algeria
{seddarinoureddine, redjimimed}@yahoo.fr

ABSTRACT: *In this paper we proposed a multi-agent model for modeling an industrial system (steam generator). Design and development of a multi-agent system (MAS) are complex problems because they require the consideration of several parts of the system which can often be tackled from different angles. The designer must identify the set of problems to solve, find multi-agent models for their resolution, implement and integrate them into a coherent system.*

Thus, the purpose of our system is to realize a virtual copy of the industrial installation of this process; this copy is intended to enable the practical training of future operators to control industrial installation. It offers an opportunity to recreate real situations, analyze control loops performances and discover the constitutions and equipment installation.

Keywords: MAS, Modeling, Simulation, Control Loops, Coherent System

Received: 10 May 2013, Revised 18 June 2013, Accepted 22 June 2013

1. Introduction

Modeling and simulation of an actual system make it possible to handle, to observe and to improve understanding of complex phenomena. Those elements are studied in different levels as for the simulation of weather changes, the physical phenomena, and systems of urban traffic.... The most of the modeled phenomena reach today complexities and high degrees of smoothness which requires the use of models and thus, the data-processing tools become more and more performing and flexible.

According to P. Fishwich's view [1], modeling is represented by symbiosis between the formalism and the techniques of modeling which follows the same target: ' to release the best metaphors and analogies to allow understanding better any phenomena.

According to C. Oussalah's view [2]: "*modeling is ear morphism between an actual system and a model that finality is to give a simplified and observable representation of the structure and behavior of the actual system*".

The preparation of the practical works may be regarded as a modeling of actual cases in laboratory. The observation of the workers carrying out their tasks by timing the times spent by in the factories carried out by Taylor, the labors of the military, the models of planes built and tested in blowers, the mathematical equations carrying of the differentials, the variables to various degree and constants which represent the actual phenomena for which the resolutions for fixed sets of entries give the solutions to these phenomena for proper defined conditions are as many examples of models which as faithfully represent as possible different realities.

However, it should be reflected in terms of accuracy, quality and cost. For example, the construction of a model of plane represents costs enough higher for a relatively low quality.

Moreover, the handling of such models is not easy and not flexible and cannot be re-used.

This concept of re-use is very significant if we want to test our model in various situations.

The data-processing tool represents a very accurate solution in several cases and its use in the field of modeling and simulation is very wide spread ensuring an excellent qualities for which the improvement is continuous for very reasonable costs.

According to Fishwick's view [3], the data-processing simulation can be understood as follows: '*Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer and analyzing the execution output*'.

Complex industrial systems are characterized by their large size, complexity of the constituent sub-systems and their dynamics, and the massive information overload. For example, the maintenance and management of complex process equipment and processes, and their integrated operation, play a crucial role in ensuring the safety of plant personnel and the environment as well as the timely delivery of quality products. Given the size, scope and complexity of the systems and interactions, it is becoming difficult for plant personnel to anticipate, diagnose and control serious abnormal events in a timely manner [4].

The goal of our work is to build a data processing simulator containing agents enable to translate a complex industrial system (Steam Generator) into a data processing program that can represents its behavior, and its interaction and verify the integrity and proper functioning of the system, it can also be used as means of mastering the complexity of these systems. Thus, we are basing on multi-agent formalism. This formalism is very used for the modeling of complicated systems.

Thanks to multi-agent systems, it is possible to model real systems in which very complex behaviors emerge from relatively simple and local interactions between many different individuals, therefore a multi-agents system is more natural to describe and simulate a system made up of entities, it allows model to appear closer to reality and it is particularly well adapted to describe a system from the standpoint of the activity of its components, that is to say when the behavior of the individuals is complex (difficult to describe with equations) [5, 6, 7]. Moreover, modeling is easier interpretable by a human observer, because description done by a MAS is more natural than simple processes; the validation by an expert will be facilitated, because he will be easily able to refer to the real world.

The rest of this article is organized as follows: The section 2 presents the problematic of our work. The section 3 presents industrial systems with their complexities in dealing multi-agent systems, which are systems based on agent approach. Section 4 presents the industrial system to simulate. The proposed multi-agent Model is described in section 5. Section 6 presente the implementation of our system. Then, we evaluate our work in Section 7. Finally, we conclude this article with a conclusion..

2. Problematic

Nowadays, the performance and quality of an industrial system is largely based on the commitment andskills of operating teams. But the industrialists do not find there a real take charge of their installations.

The current methods of training are based on the theoretical approaches; the apprentices can disturb the instructions or the other operators during the training. The interview of the behavior of the operator is done apart from the situation of operation of the process. This can lead to cases where there is impossible to build up competences of the drivers' installations.

So it is difficult with the current methods of training, to train skilled operators able to control industrial installations. Use a simulator to provide trainings can solve such thorny problems. It allows us to provide trainings outside the site in a favorable context by reconstituting many scenarios without damage risk, without disturbance of the instructions and an easy evaluation of operators' skills.

In our work, we used the multi-agents approach rather than the oriented-object approach, because the latter does not provide adequate concepts and mechanisms to model the complex systems in which the reports dynamically evolve such as this kind of system (vapor generation process using a water station). Moreover, the objects are generally passive whereas the agents are permanent and active and autonomous they are also responsible for their actions whereas the objects are not always responsible for their actions. Thus multi-agents simulation appears to us being the most adequate approach with the modeling of this kind of system. Multi-agents Simulation is a recent field of research, which is based on the concept of agent.

3. Complex Industrial Systems Multi-Agent Systems (MAS)

Modern companies are based on very complex industrial systems and the control of information technologies has enabled to design, develop and implement. One of the main reasons for the complexity of industrial systems is they are often themselves systems of systems, and they are therefore the result of the integration of many other industrial systems relatively complex.

The multi-agent systems (MAS) are now widely used for the modeling of such systems and lead most of the time to series of exploratory simulation, which simplifies the construction, maintenance and implementation of complex systems.

3.1 The complexity of an industrial system

An industrial system is often classified as a complex system when the decomposition hierarchy of such a system reveals a large number of sub-systems or sub-systems of a very different nature. The two main sources of complexity of an industrial system are indeed the size, heterogeneity and the large numbers of interactions, these parameters are also most often present simultaneously in a complex industrial system.

The main difficulty in the realization of a complex industrial system is of course the need to manage effectively the complexity of two parameters that we have just enumerated. This is the field of systems integration, that is to say, the ability to produce a new industrial system from several existing systems. Integrate a system fundamentally consist in inter-operate many heterogeneous components of this system in order to obtain a homogeneous overall behavior (in relation to its environment) of the system resulting from this process.

A complex industrial system is also a system that is characterized by the difficulty to make subsystems compose it cooperate homogeneously (highly heterogeneous or in great number).

3.2 Multi-Agent Systems (MAS)

Multi-agent systems are an emerging conceptual paradigm to simulate the interactionof multiple autonomous agents in an environment [8, 9]. Multi-agent systems havemany applications; our interest is in their use to build operational simulator of an industrial system. In general, a system is called multi-agent if the system contains at least one agent that perceives a simulatedenvironment through its actions influence the environment and areinfluenced by the perceived situation in the environment.

3.2.1 What is a Multi-agent System?

According to Ferber's view [10], the term '*multi-agent system*' refers to a system consisting of the following parts:

• The environment E consisting of the following elements:

- A set of objects O. Objects can be perceived, created, destroyed and modified by agents.
- A set of agents A. Agents are a subset of objects ($A \subseteq O$) capable of performing actions - the active entities of the system.
- A set of locations L determining the possible position of the objects (from the set O) in space.

• An assembly of relations R which link objects and also agents to each other.

• A set of operations Op enabling the possibility for agents to perceive, manipulate, create, destroy objects of O, in particular representing the agents' actions.

• A set of operators U with the task of representing the application of the operations from Op and the reactions of the world to this attempt of modification. The operators from U are called the laws of the universe.

3.2.2 Definition of an Agent
According to the heterogeneity of the field there is no common agreement about a definition of the term agent. One primary characteristic that differentiate agents from an ordinary program is that the agent must be autonomous. Several definitions of agents includes this characteristic, for example:

• "*Most often, when people use the term 'agent' they refer to an entity that functions continuously and autonomously in an environment in which other processes take place and other agents exist* [11]."

• "*An agent is an entity that senses its environment and acts upon it* [12]."

• "*An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, in pursuit of its own agenda and so as to effect what it senses in the future* [13]."

Although not stated explicitly, Russell's definition implies the notion of autonomy as the agent will act in response to perceiving changes in the environment. The other four definitions explicitly state autonomy. But all definitions add some other characteristics, among which interaction with the environment is mentioned by most. Another identified feature is the property of the agent to perform specific tasks on behalf of the user, coming thus to the original sense of the word agent, namely someone acting on behalf of someone else.

One of the most comprehensive definitions of agents, is the one given by Wooldridge and Jennings view [14] in which an agent is:

• "*A hardware or (more usually) a software-based computer system that enjoys the following properties: autonomy - agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state; social ability - agents interact with other agents (and possibly humans) via some kind of agent-communication language; reactivity: agents perceive their environment and respond in a timely fashion to changes that occur in it; pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking initiative*."

3.2.3 The Environment
Common to all environments is that they provide percepts to the agent and that the agent performs actions in them. Multi-agent theory regards the environment as an integral part of the framework. In general, two classes of environments can be distinguished: artificial and real environments. Agents that are computer programs and exist in artificial software environments are called software agents.

Russell and Norvig discuss a number of key properties of environments that are now adopted by most researchers in the domain:

Accessible versus inaccessible: indicates whether the agents have access to the complete state of the environment or not.

• **Deterministic versus nondeterministic:** indicates whether a state change of the environment is uniquely determined by its current state and the actions selected by the agents or not.

• **Static versus dynamic:** indicates whether the environment can change while an agent deliberates or not.

• **Discrete versus continuous:** indicates whether the number of percepts and actions are limited or not.The most complex class of environments are those that are inaccessible, nondeterministic, dynamic and continuous. The first three properties of this list are properties typically occurring in MASs [15].

4. Global description of the system to be simulated

In this section, we will show the architecture and the operation of the process concerning steam generator.

4.1The feed water station
The feed water station is a method that feeds the steam generator, it is constituted by:

• The condenser drunk. (15E01)

• The degasser (15B02

• The feed water tank (15B01)

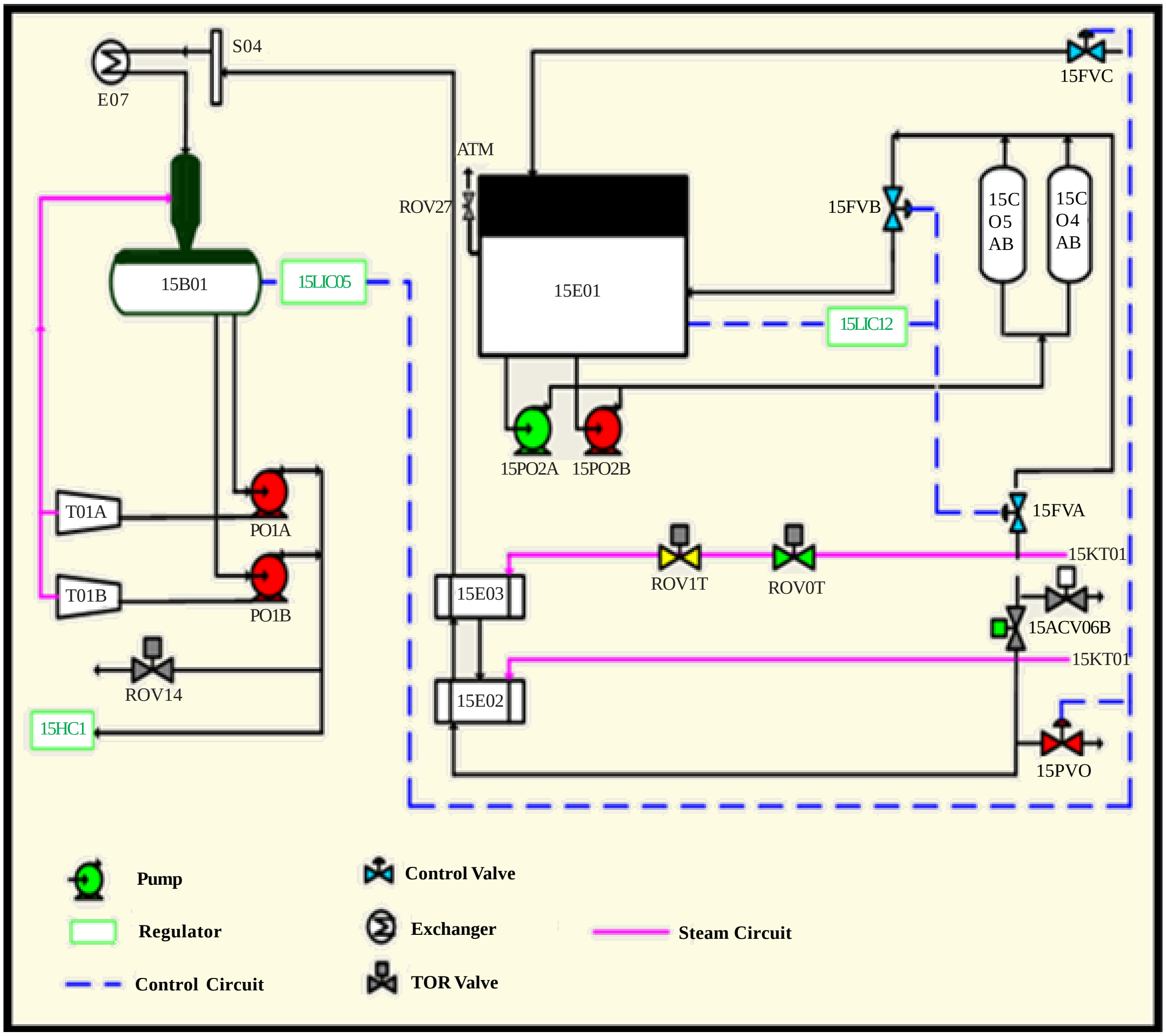

Figure 1. Technical drawing of feed water station

4.1.1 Circuit description

The extraction water comes from 15 E 01 by the bottom, it gets through a valve and a mesh filter before its inhalation by the pumps 15 PM 02 A/B.

The water is inhibited with 12 bars towards the maintaining ejectors 15 E 05 A/B, and it will feed the heaters while passing by 15 FVA towards the degasser and the 15FVB returns towards the condenser. Those two valves functions are in split range and controlled by15 LIC12 (level 15 E01).

Then, the water crosses 15 ACV 06 A/B (A towards the heaters, B towards the channel rejection).

The 15 FVD goes towards U50 and 15 FVA supplement of Unit 50 towards the 15 E 01, and are controlled in split range by the 15 LIC 05 in order to maintain a constant level in the feed water tank .

The extraction water is then conducted towards 15 E 02 and 03 in contact of steam racking 2 and 3 of the 15 KT 01. The two heaters are provided with By-pass, the extraction water goes then to feed the degasser.

4.1.2 Regulations ensuring the feeding of water

The level of water in the condenser (15E01) regulated to 65% by a loop of regulation 15 LIC 12.

The level of water in the feeding tank (15B01) regulated to 65% by a loop of regulation 15 LIC05.

4.2 The steam generator

A boiler is a steam generator; it aims to raise the temperature of water until the change of its status, that is to say, to become steam, and then to bring it to proper defined pressure and temperature.

In the construction of a boiler we distinguish:

- A metal frame with its trim, masonry casing.
- The boiler: tubes, box back, screens, spray, super heaters, balloons…
- The combustion chamber: burners, air and auxiliary circuits.

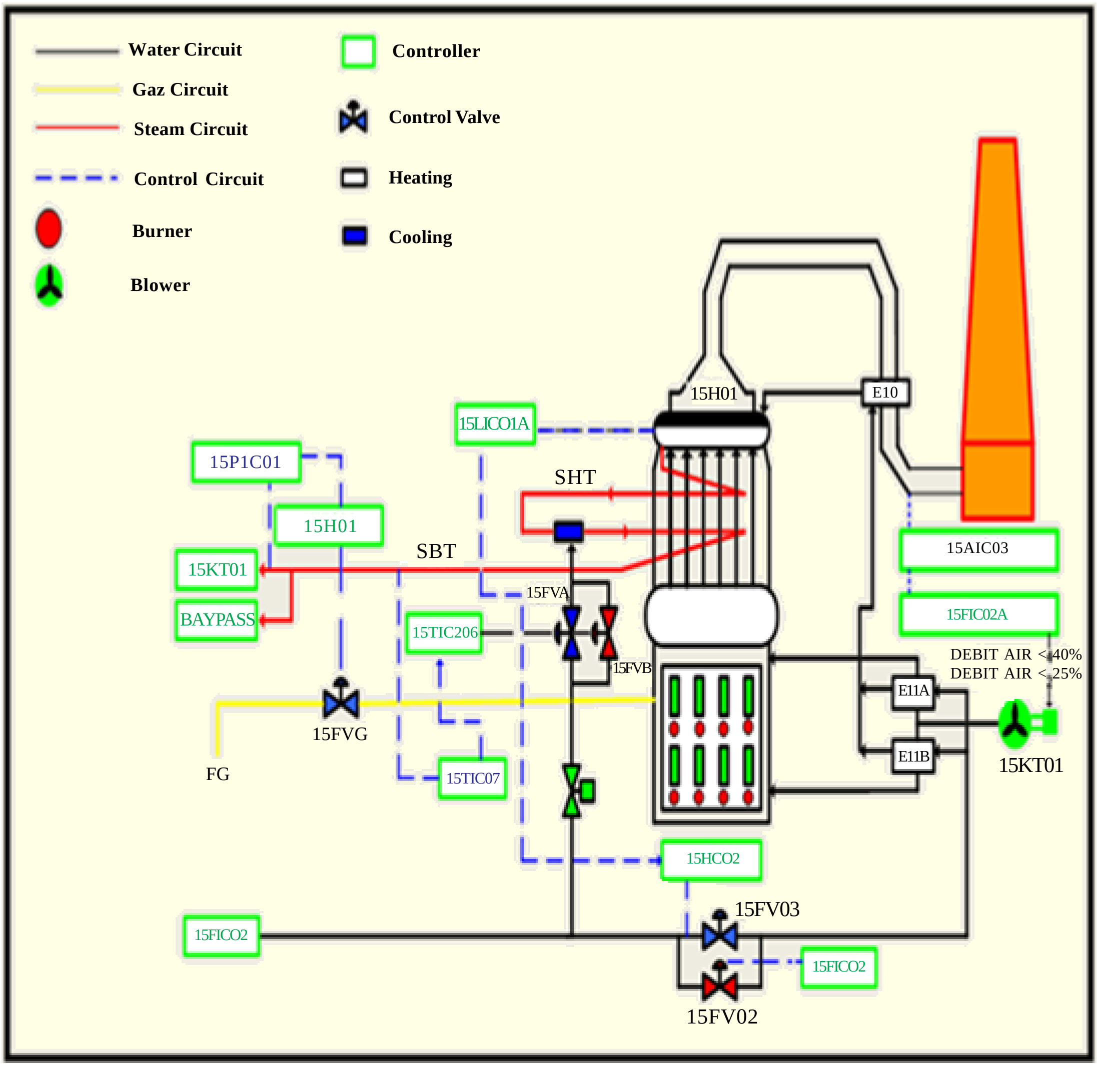

Figure 2. Technical drawing of the steam generator

4.2.1 Principle of operation

We said that a boiler is a steam generator for producing dry steam at a determined pressure and temperature.

To obtain this vapor, we have to burned a fuel in the boiler which can be:

• A solid fuel (coal ,wood ,peat ...).

• A liquid fuel (made from petroleum, oil, coal tar…).

• A gaseous fuel (gas down, blast furnace gas, refinery gas, natural gas or as in our case).

The first boilers were simple tight container which is placed in the furnace. Heat is transmitted and steam is produced, were obviously in direct relation with the heated surface. Attempts have been made to increase it by breaking down the surface in elements which we give a circular section to remain under pressure, and a small diameter to increase the surface achievable in a given volume.

The fuel is burned in an ability called furnace. This ability can be internal to the boiler (cylindrical marine boiler, locomotion boiler). It is generally external especially in water-tube boilers and limited to a number of these surfaces by refractory masonry.

The gases produced by combustion pass through the boiler and yield part of their heat to the walls, causing the evaporation of the water. At the outlet side of the boiler there is a chimney through which they are discharged into the atmosphere. The difference in density between hot gases of chimney and the cold outside air causes the raising of the gas and ensures the drawing which drives the air into the furnace.

In extensive installations, we activate the circulation by a fan, or even by two fans one forcing air into the furnace (fan inflation), the other sucking fumes (induced draft fan).

The water is brought to the boiler by a feeding-water pump. The power of the boiler is characterized by the intensity of steam expressed in kg produced per hour.

a) **Combustion air circuit:** The air is provided by two blowers fans, the first one is conducted by turning turbine and the second one operates through an engine (those two fans do not operate simultaneously), the air is then heated by the air super- heater before entering to the burners by using the heat of evacuated gases. The air flow is regulated according to the fuel to have a complete combustion.

b) **Water circuit:**The feeding of water is done starting from a pump controlled by two valves mounted in parallels, the first one is used to fill the balloon and the other one for regulation purpose (body of adjustment/regulation); from where it comes out to go then by an economizer in order to be heated by the calories of the evacuated smoke and returns finally to the higher balloon to produce the steam.

c) **Steam circuit:** The saturated steams comes out from the higher balloon, goes by the super heater (in the hearth), its temperature increases while its pressure varies slightly. At the end, the final temperature of the produced steam will be regulated by the desheater (water injection).

This dry steam is also called load of the boiler.

4.2.2 Regulations which ensure the generation of steam

Five conditions are required to ensure a proper production of steam:

1. Regulation of heating (a correct air-gas factor).
2. Control of Combustible flow (15FVG).
3. Control of air flow (15KT01).
4. Regulation of balloon level (15H01) to 50%.
5. Control of superheated steam temperature (495 C°).

5. Proposal for Multi-Agent model of simulator

To model our system we have used the organizational model Aalaadin [16], which constitutes the basis of our design.

The organizational model Aalaadin is a project that focuses on the analysis, design, formalization and implementation of multi-agent systems in an organizational perspective. The underlying conceptual model (AGR) is based on the concepts of agents, groups and roles [17].

5.1 AALAADIN model

With AALAADIN one can describe multi agent systems with different forms of organizations such as market-like or hierarchical organizations and therefore it could be useful for designing MAS. An organization in AALAADIN is a framework for activity and interaction through the definition of groups, roles and their relationships. Figure 1 represents the diagram of this model.

5.1.1 Agent

An agent is only specified as an active communicating entity which plays roles within groups. The model places no constraints on the internal architecture of the agents. Figure 3 represents the Agent model.

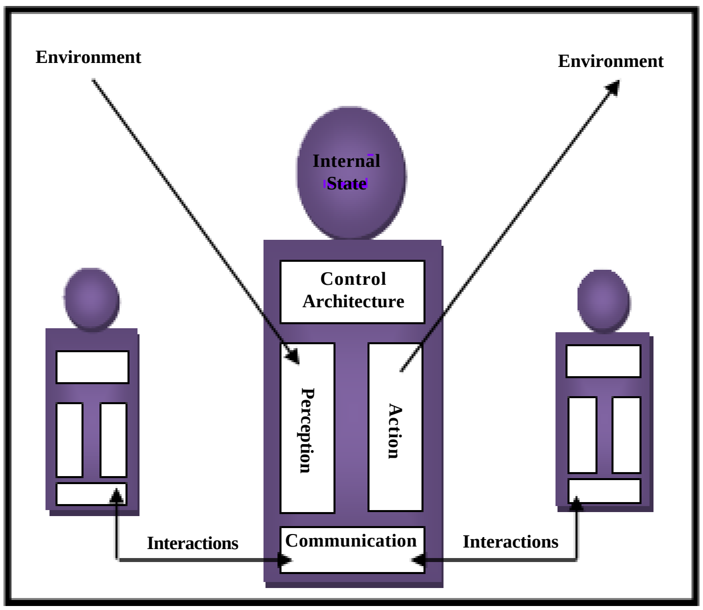

Figure 3. Agent model

5.1.2 Group

Groups are defined as atomic sets of agent aggregation representing any usual multi-agent system. Each agent is part of one or more group. In AALAADIN groups can freely overlap each other.

5.1.3 Role

The role is abstract representation of an agent function, service or identification within a group. Each agent can handle multiple roles, and each role handled by an agent is local to a group.Figure 4 represents the diagram of AALAADIN mode.

Therefore, we have split up our process into five groups each groups contains a set of the agents. The following diagram represents The organizational structure of the simulator based on the proposed AGR model.

5.2 Groups and Roles of simulator

5.2.1 Perturbation group

This group contains

5.2.1.1 The APM Agent

The agent system is a principal agent; it is the meeting point of the system and the external environment. It periodically sends orders to agents and groups of system until the adjustment of measurements; it ensures the connection between these groups thanks to these different communications.

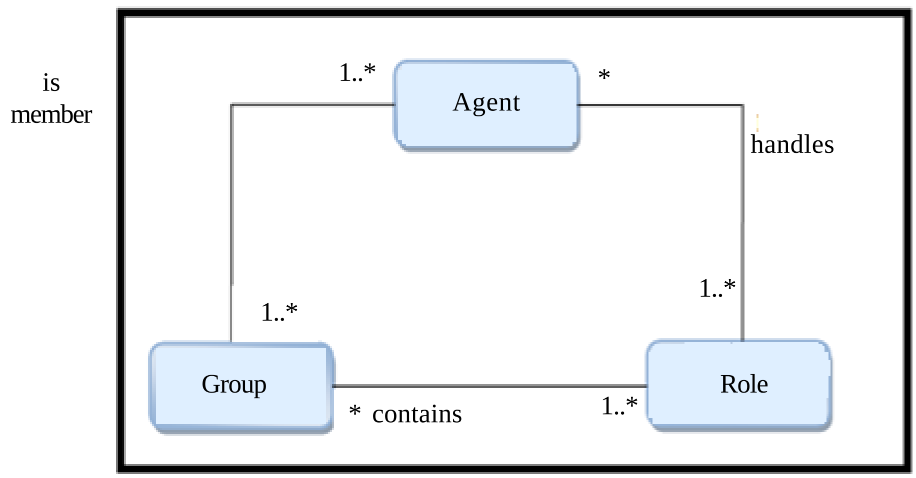

Figure 4. Agent/Group/Role model

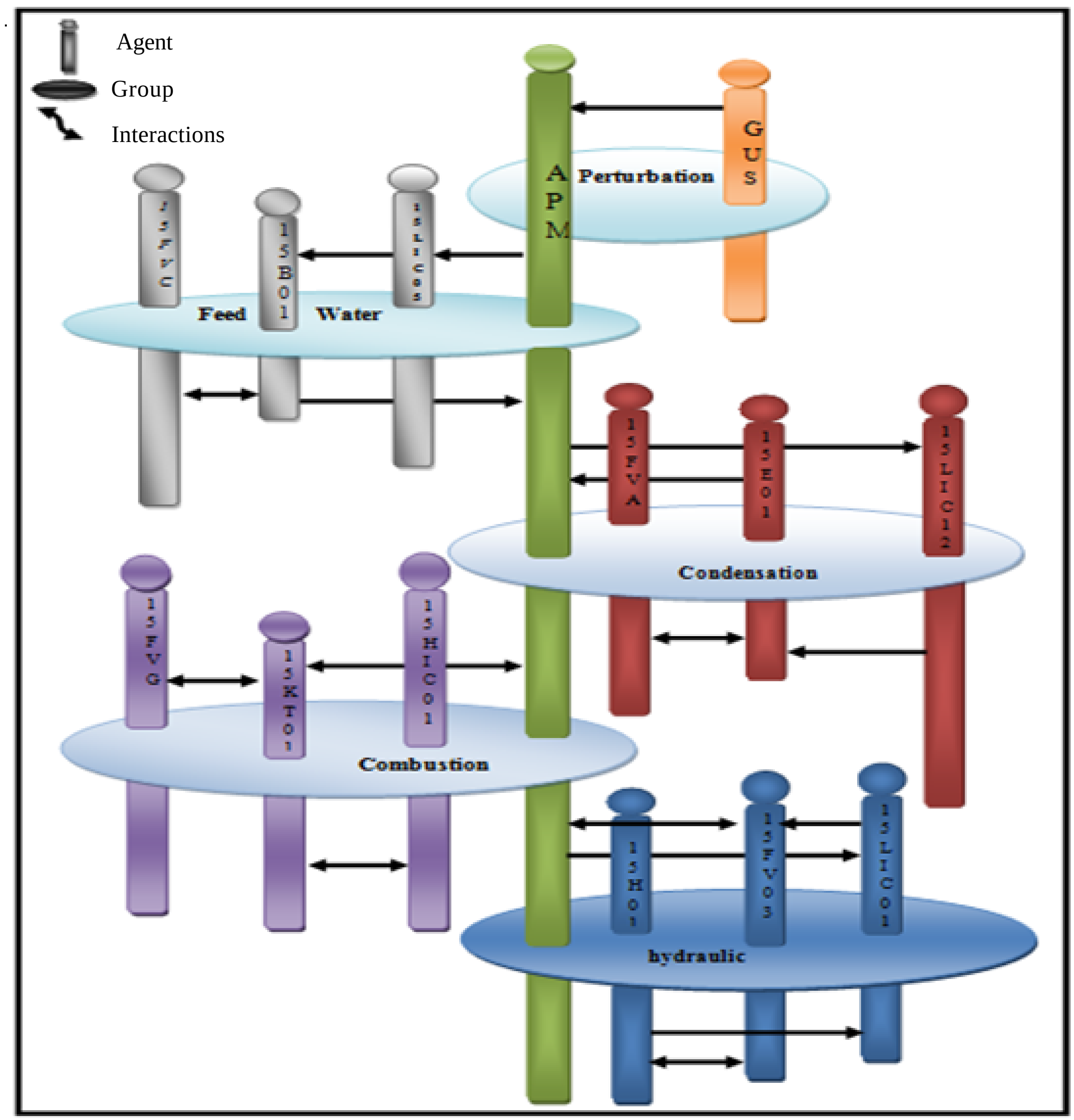

Figure 5.Organizational structure of the simulator

APM Agent belongs to all groups of the simulator.

5.2.1.2 The GUS Agent

Its role is to trigger the disturbances; thanks to this agent the operator controls the system thanks to instructions of the load of steam generator on this Agent system.

The disturbing agent communicates with the Agent system by sending the instructions.

5.2.2 The feed water tank group

This group contains a set of agent in interaction ensures the operation of water supply of the boiler.

5.2.2.1 Agent (15LIC05)

Is in charge of the level of the tank, its role relates to the regulation of the tank, by using the necessary values, provided by the other agents of the system, the Alimentation agent (15LIC05) sends the orders of opening and closing thanks to the agent Valve-tank (15FVC) in order to modify the flow of water therefore modifies the level of the tank (15B01).

5.2.2.2 Agent (15B01)

It represents the variation of the level of water inside the tank, it communicates with the Alimentation agent, by periodically sending the level in order to be regulated, and on the other hand it receives orders of the Alimentation agent (15LIC05).

5.2.2.3 Agent (15FVC)

This agent controls the flow of water entering the condenser(15E01), it is in communication with the tank agent (15B01) and condensation agent (15LIC12) by providing the flow of water.

5.2.3 The condensation group

This group contains a set of agent in interaction in order to execute the operation of water condensation.

5.2.3.1 Agent (15LIC12)

It is the responsible for the level of the condenser (15E01), its role relates to the regulation of the condenser, by using the values provided by the other agents of the system, the Condensation agent (15LIC12) sends the orders of opening and closing using the condenser valve agent (15FVA) in order to modify the flow of water and to modify the level of the condenser.

5.2.3.2 Agent (15E01)

It represents the variation of the level of water in the condenser, it communicates with the Condensation agent (15LIC12) by sending this level periodically for which is regulated on the other hand it receives orders of the Condensation agent.

5.2.3.3 Agent (15FVA)

This agent controls the flow of water flowing out to the tank (15B01); it is in communication with the Condenser agent (15E01), the Condensation agent(15LIC12) and the Alimentation agent (15LIC05) by providing the flow of water.

5.2.4 The combustion group

This group contains a set of agents in interaction making possible to execute the operation of combustion.

5.2.4.1 Agent (15HIC01)

The role of this agent is to control the combustion and to regulate the measured sizes which it constantly receives from various agents of system. Indeed, if these measurements are not compatible with the load of the boiler it periodically sends orders of regulation towards the Ventilator agent (15KT01) and the Gas valve agent that is to say agent (15FVG) in order to reach desired sizes.

5.2.4.2 Agent (15KT01)

Its role is to control the air flow; it periodically sends this flow towards the combustion agent (15HIC01) and receives the orders coming from this agent.

5.2.4.3 Agent (15FVG)

This agent controls the gas flow; it periodically sends this flow towards the combustion agent (15HIC01) which in its turntransmits orders to it (opening, closing) in order to modify the gas output.

5.2.5 The hydraulic group

This group contains

5.2.5.1 Agent (15LIC01A)

Its role is to regulate the level of the balloon by using values provided by the various agents of the system, the Hydraulic agent sends the orders of opening and closing towards the water valve agent that is to say agent (15FV03) till this measured level become equal at the desired level (50%).

5.2.2.2 Agent (15FV03)

The role of this agent is to control the flow of feedwater tank coming from feed water station, it periodically sends this flow towards the agent balloon (15H01) and the hydraulic agent (15LIC01A) and receives the orders (opening, closing) coming from the latter.

5.2.2.3 Agent (15H01)

Its role is to calculate the level of balloon by the use of the sizes provided by APM agent and the water valve agent (15FV03); it periodically sends this level towards the hydraulic agent (15LIC01A) to be controlled.

6. Implementation of the system

We will develop our "*multi-agent simulator*," using Mad Kit agent platform because this platform is based on Aalaadin model. The MADKIT platform is built around AALAADIN model. In addition to three core concepts, the platform adds three design principles:

- Micro-kernel architecture
- Agentification of services
- Graphic component model

MadKit Itself is a set of packages of Java classes that implements the agent kernel, the various libraries of messages, probes and agents. It also includes a graphical development environment and standard agent models. But the platform is not an agent platform in the classical sense as any service besides those assured by micro-kernel is handled by agents.

The reduced size of micro-kernel, combined with the principle of modular services managed by agents enable a range of multiple and scalable platforms.

6.1 Agent micro-kernel

MADKIT micro-kernel is a small and optimized agent kernel. The kernel itself is wrapped in a special agent KernelAgent which permits control and monitoring of the kernel within the agent model. Kernel only handles the following tasks:

6.1.1 Control of local groups and roles

The micro-kernel is responsible for maintaining correct information about group members and roles handled. It also checks if requests made on groups and roles structures are

6.1.2 Agent life-cycle management

The kernel is responsible for the creation and destruction of agents and maintaining tables and references of agent instances. It assigns a global unique identifier AgentAddress (kernel address plus identification of agent on local kernel) to each agent upon its creation.

6.1.3 Local message passing

The kernel manages routing and distribution of messages between local agents. The basic mechanism relies on a copy-on-write implementation to avoid unnecessary operations.

6.1.4 Kernel hooks

The kernel itself is fully extensible through kernel hooks, which allows extension of the core behavior of the platform. Any agent can request KernelAgent to subscribe to the kernel hooks.

6.2 Agents

Group and roles actions and requests are defined at action level. The agent developer is completely free to define the agent behavior, but the organizational view will be always present.

6.2.1 Control and life-cycle

The main agent class in MADKIT defines primitives related to message passing, plus group and role management, but does not implement a specific execution policy. A subclass adds support for concurrent, thread-based execution, which is the natural model for coarse-grained collaborative or cognitive agent. Additional subclasses implements synchronous execution through an external scheduler, focused on reactive or hybrid architectures: many fine-grained agents.

6.2.2 Communication

It is achieved through asynchronous message passing, with primitives to directly send a message to another agent represented by its AgentAddress, or the higher-level version that send or broadcast to one or all agents having a given role in a specific group, exemple:

```
void nouveau_débit_eau (int eau)
{    débit_eau = eau;
Grand mes =new Grand ();
mes.setnature ("débit de l'eau");
mes.setv1 (débit_eau);
ObjectMessage obj = new ObjectMessage ((Object) mes);
AgentAddress adr = getAgentWithRole ("15LIC01A", "APM ");
sendMessage(adr, obj);
}
```

In this exemple the 15LIC01A agent sends the new water flow towards APM agent:

6.2.3 Message passing

Messages in the MADKIT platform are defined by inheritance from a generic Message class. Thus specific messages can be defined for intra-group communication, and allows a group to have its specific communication attributes. Messages receivers and senders are identified with their *AgentAddress*. MadKit do not define interaction mechanism, which can be defined on an ad-hoc basis, or built in a specific agent model library.

6.3 Agentification of services

MADKIT uses agents to achieve things like distributed message passing, migration control, dynamic security, and other aspect of system management. These different services are represented in the platform as roles in some specific groups, defined in an abstract organizational structure. This allows a very high level of customization, as these service agents can be replaced without hurdle.

6.3.1 Communication and distribution

As messaging, as well as groups and roles management uses the 1) *AgentAddress* identifier, and as this identifier is unique across distant kernels, MADKIT agents can be transparently distributed without changing anything in the agent code. Groups can spawn across different kernels, and agents usually do not notice it. Distribution in the agent platform relies on two roles in the system group:

6.4 Componential graphical architecture

MADKIT graphic model is based on independent graphic components, using the Java Beans specification in the standard version. Each agent is solely responsible for its own graphical interface, in every aspects (rendering, event processing, actions...) An agent interface can be a simple label, a complex construction based on multiple widgets, or a legacy bean software module. A "*graphic shell*" launches the kernel and setup the interfaces for the various agents and manage them in a global GUI (for instance: each agent has its own window, or is mapped in a global worksheet, or is combined with another agent interface,...). As the graphic shell is a classic software module, it can be wrapped in an agent for maximum flexibility, allowing control of other agent interfaces by a regular MADKIT agent that can be part of any interaction scenario. [18]

We will present two main interfaces of our project, the interface of the steam generator section and the interface of feedwater station section they allow the user to drive the regeneration steam operation and supply boiler with demineralized water by publishing guidance to various regulatory bodies and follow the operation of the process functioning as shown by the following figures:

Figure 6. Interface of simulator boiler section

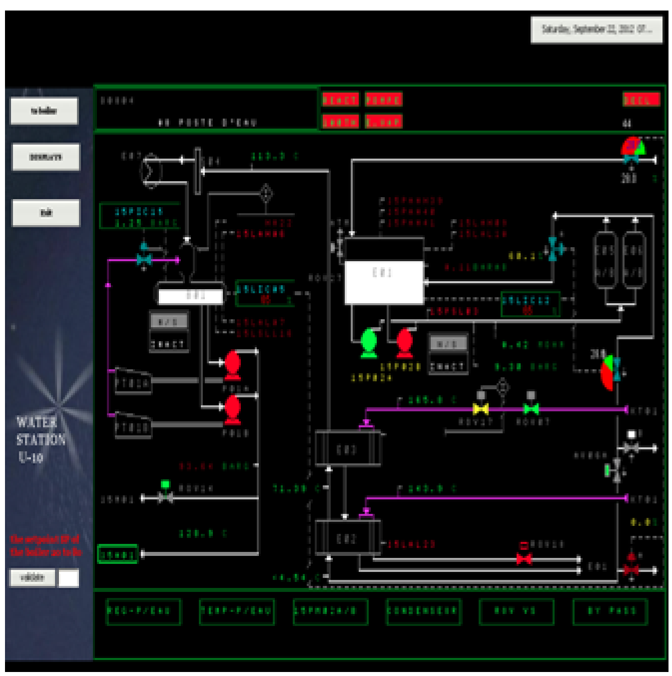

Figure 7. Interface of simulator water station section

7. Evaluation

In our system we used the Mad Kit agent platform [5] which is based on the organizational model Aalaadin. with this model, organizational concepts such as groups, roles, structures, outbuildings …, are considered as the building blocks allow the development of heterogeneous systems, and also allow to model and simulate multi-agent systems.

So our work can be considered as an assistant of training and learning for beginner operators, workers new recruits and trainees enabling them to understand and simulate the steam generator process with the feed water station without acting on the real system and without stopping processes and generate losses of money and time. So we can consider our simulator more as a teaching aid explaining the Multi-Agents approach and treats one of its domains of use which is the simulation of systems and more particularly complex industrial systems.

8. Conclusion

A data-processing model is the representation and the implementation of a system on a computer. It simulates phenomenon using the computational capabilities of the computer.

Among data processing models, there are models based on multi-agent systems (MAS). In a MAS modeling, the phenomenon is described from the perspective of its constituents. The aim is to describe the global behavior having a representation of the actions of a set of autonomous and active entities.

Furthermore, oriented agent simulation is one of the most recent techniques of simulation for complex systems such as biological and industrial systems, It takes a very important place and shows good success in this domain.

This work allowed us to operate and improve our knowledge in industrial and in the modeling and simulation of complex systems, and is intended for the training of operators to control the installation of the steam generator and feed water station on site. It can be considered as a tool to learn and understand the functioning of the machine in reality for all involved.

References

[1] Fishwick, P. (1995). Simulation Model Design and Execution : *Building Digital Worlds*.

[2] Oussalah, C. (1988). Modèles hiérachisés multi-vues pour le support de raisonnemen Dans les domaines techniques. *Technical report*.

[3] Fishwick, P., Paul, A. (1997). Computer simulation: growth hrough extension. *Transactions of the Society for Computer Simulation International*, 14 (1) 13–23.

[4] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., Yin, K. (2003). A review of process fault detection and diagnosis part 1, 2, 3, *Computer & Chemical Engineering*, 27 (3) 293–346.

[5] Chaib-Draa, B. (1999). Agent et Système Multi - Agents. Université Laval, Quebec (Canada), November.

[6] Ferber, J. (1995). Les systèmes multi-agents : vers une intelligence collective. *Informatique, intelligence Artificielle Interreditions Paris*.

[7] Krishnamurthy, E. V., Murthy, V. K. (2006). Distributed agent paradigm for soft and hard computation. *Journal of Network and Computer Applications*, 29 (2) 124–146.

[8] Weiss, G. (ed.). (1999). Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence. *MIT Press, Cambridge, MA*.

[9] Wooldridge, M., Weiss (ed.), G. (1999). Multiagent Systems – A ModernApproach to Distributed Artificial Intelligence, *MIT Press, Cambridge, MA*, p. 27–77.

[10] Ferber, J. Multi-Agent Systems. (1999). An Introduction to Distributed Artificial Intelligence. Addison-Wesley.

[11] Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60, p.51-92.

[12] Russell, S. J. (1997). Rationality and intelligence. *Artificial Intelligence*, 94, p.57-77.

[13] Franklin, S., Gasser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. *In*: Muller, Wooldridge, and Jennings, eds. Intelligent Agents III. Agent Theories, Architectures, and Languages. Springer Verlag, p.21-35

[14] Wooldridge, M., Jennings, N. R. (1995). Agent theories, architectures, and languages. *In*: Wooldridge and Jennings, eds. Intelligent Agents, Springer Verlag, p.1-22.

[15] Russell, S., Norvig, P. (2003). Artificial Intelligence: A Modern Approach, Prentice Hall.

[16] Ferber, J., Gutknecht, O. (1998). Aalaadin: a meta-model for the analysis and design of organizations in multi-agent systems, ICMAS (International Conference on Multi-Agent Systems), Paris, Y. Demazeau (ed), *IEEE Press*, p. 128-135.

[17] Gutknecht, O., Ferber, J. (2000). The MadKit agent plateforme architecture. Laboratoire d'Informatique, Robotique et Microélectroniquede Montpellier.

[18] Fabien Michel, Jacques Ferber, Olivier Gutknecht. (2001). Generic Simulation Tools Based on MAS Organization. LIRMM Laboratoire d'Informatique, Robotique et Micro-´electronique de Montpellier.