JADE Modeling for Generic Microgrids

Guillaume Guerard and Hugo Pousseur

Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France,
`FirstName.LastName@devinci.fr`

Abstract. Around the world, smart grids are being developed to reduce the electric waste and to prevent blackouts. Simulating microgrid, an eco-district or virtual power plant, is challenging, considering their different behaviors and structures. Each one varies according to several aspects: social, economic, energetic, mobility and the well-being of its inhabitants. This paper proposes a demand-side management of a microgrid with a systemic approach, the model is based on the JADE framework and generic data from the literature. This paper focuses on the ability of microgrid to regulate its consumption with flexibility.

Keywords: Microgrid, Demand-side management, Multi-agent system, JADE

1 Introduction

The energy development has resulted in a change of paradigm in the 21st century. Industrial and political entities, practitioners and the scientific community are seeking for smarter cities [1]. The French Institute for Demographic Studies (INED) declares the world population will increase to 10 billion inhabitants at year 2050, while the urban population will double (increase by 63%). The extended urbanization requires new ways to understand and manage the complexity of the energy production and consumption, while the world tends to over-digitalize, to exploit big data and to disrupt classic transaction through blockchains.

Nowadays, although the cities occupy only 2% of the planet's surface, they accommodate about 50% of the world population, consume 75% of the total generated energy, and are responsible for 80% of the greenhouse effect[1]. The development of smart cities is dependent on the level of intelligence of electrical networks, from the producers to the consumers, from the consumers to producers. The most important aspect is the coordination between all entities; a microgrid will be able to stimulate the consumers to modify their consumption in critical conditions to maintain the electrical infrastructure unaffected.

A microgrid contains various entities. Consumers are prosumers, i.e. they consume energy and have various small, modular generation systems together

[1] UN Environment Program, Visions for Change. Recommendations for Effective Policies on Sustainable Lifestyle, 2011

with storage devices. Such systems can be operated interconnected to the grid or islanded [1, 2, 3].

Smart grids and smart cities need to be understood. They are commonly described as complex system and the better way to analyze it is by modeling and simulating [2]. In this context, the goal is to create a context-free model in order to give a decision making tools for social, economic and algorithmic questions about a smarter microgrid.

This paper is divided into four sections. The second section presents the JADE architecture and the microgrid's model. The third section presents cases studies. The fourth section concludes this paper.

2 Model

In this paper, the model is being developed on JADE[2] (Java Agent DEvelopment Framework), a software framework to built and to develop multi-agent applications [4].

There are several definitions to define multi-agent system. However the definitions have common features. A multi-agent is defined as a system grouping several agents in the same environment. Each agent can interact with its environment and other agents, in order to solve a complex problem. They divide the problem to a number of smaller and simpler ones that can be solved in more efficient computational ways than using a single-agent system [5].

2.1 Simulation model

Each simulated device is defined by an agent. This agent is qualified as reactive agent, it waits an order to be launched. Moreover devices are divided into two different categories [6] as follows:

- **Cyclic device:** devices which the end task is known at launch, and the process is cyclic. For example: washing machine.
- **Acyclic device:** devices which the end task can't be known. For example: Heating, Ventilation and Air-Conditioning.

To enhance the quality of simulation, the running time of acyclic device is controlled randomly. Furthermore, device gets some features:

- **Daily frequency:** the number of uses per day .
- **Time slot:** storing all interval times which the device has to been launched, based on the french national statistics agency values (Gaussian distribution over these values).
- **Operating modes:** list of modes, each mode is delimited by a minimum and maximum power consumption which the operating time is proportional and linear. It's possible for each mode to choose a power between its minimum

[2] Official website: http://jade.tilab.com/

and its maximum, this decision will have an impact on the execution time. For example, the washing machine gets several modes: cleaning, spinning and rinsing.

In each smart house, an agent called *controller* stores all plugged devices and computes the best strategy. This part is centralized by choice, for raising the realism, a decentralized system imply each device gets artificial intelligence and the same way of thinking. Moreover, to communicate with other houses, a *home* agent has a gateway to create a link between all *controller* agent. These agents are qualified as cognitive agent.

This division of tasks allows to lighten the *controller* agent and raise the security as a fence. having an *home* agent helps to improve the cyber-security of a home, because the *controller* agent is not directly on the home networks.

By contrast, *home* agent participates in a decentralized system. Based on the process of leader-follower, its goal is to discuss with other homes and find the best strategy according to the amount of available energy. If the system was centralized by a server, in the case where the server shutdown for any reasons, the system is broken. In the presented model, if a home is disconnected from the network, the system continues to run.

Each home is stored in an individual platform, JADE attributes a MTP address, allowing other platforms to communicate with it. In the case of large simulation, it's possible to distribute homes on several computers/servers.

An interface allows the user to configure the generic microgrid:

- Goal consumption by default, until the new order.
- Duration of one day simulation.
- Number of connected smart houses.
- Time frequency for computing the new strategy.
- Device's number interval range.
- Use or not of global strategy.
- Duration of timeout control.

During the simulation, the *controller* agent and *home* agent have to take some decisions.

2.2 Agent's strategies

The *controller* agent has a list of devices, with static information about devices as daily frequency and dynamic information as the current state (on,off). With these information and the goal's consumption, the *controller* agent seeks for the best combination by solving a 0-1 knapsack problem. Element gets a value and a weight, the algorithm searches the best combination of elements, maximizing the sum score and respecting the total weight constraint. A knapsack problem is modeled as follows [7]:

> Several objects are available (n), each object get two characteristic, a weight and a value. The knapsack is limited by a weight maximum (M). The goal is to find the best combination in order to obtain the maximum value (1) in terms of the weight imposed (2).

$$\text{maximize} \sum_{i=0}^{n} x_i * s_i \tag{1}$$

$$\text{subject to} \sum_{i=0}^{n} x_i * c_i \leq M \text{ and } x_i \in \{0,1\} \tag{2}$$

In this context, the weight corresponds to the consumption. The literature about the knapsack problem have a large amount of application on consumption's scheduling. It is known that the knapsack problem is a useful tool to smooth the consumption curve under a value or to manage the energy distribution amoung various devices [8, 9, 10]

Each device is represented by a couple of consumption and score. To attribute a score, the *controller* agent updates regularly (in discrete time) a new value for each device based on (3).

$$score = \begin{array}{l} \mathbb{1}_{time\_of\_use} * \mathbb{1}_{state\_of\_use} \\ *\|frequency\| * \|consumption\| \end{array} \tag{3}$$

$$\mathbb{1}_{time\_of\_use} = \begin{cases} 1 \text{ can be launched at the current time} \\ 0 \text{ else} \end{cases} \tag{4}$$

$$\mathbb{1}_{state\_of\_use} = \begin{cases} 1 \text{ current state is off} \\ 0 \text{ else} \end{cases} \tag{5}$$

$$\|frequency\| = |\text{already time used} - \text{frequency per day}| \tag{6}$$

$$\|consumption\| = \begin{cases} \text{(consumption available -} \\ \text{average consumption device)} \\ 0 \end{cases} \tag{7}$$

The indicator functions (4) and (5) allow to verify the possibility to launch a device. If the device is already launched (5) or it's not able to use at the current time (4), the value is equal to zero, in this case the device can't be launched.

The frequency norm allows quantifying the usefulness to launch a device depending on the number of uses per day (6) and the consumption (7). For example, the hot water tank is commonly heated two times per day during the night and after the morning. Once the hot water tank is used two times in the simulated day, it doesn't need energy until the next simulated day. The function (7) checks if there is more than enough energy unused to launch the device. Due to these functions a value is attributed (3) and regularly updated for each device.

During the process, the *home* agent discuss with (n) other smart houses to communicate its consumption (c_i) in order to regulate the global consumption depending to the amount of available energy (GC), i.e $\sum_{i=0}^{n} c_i = GC$.

This situation can be compared to a consensus problem [11] where the smart houses form a graph communicating together to solve the same problem. The process is based on leader-follower allowing temporarily attribute a home as the current leader taking the decision for the neighborhood. The leader computes the new goal consumption of each home (g_i) as follows: all homes apply the same percentage on their current consumption

$$g_i = (2 - \frac{\sum_{j=0}^{n} c_j}{GC}) * c_i \tag{8}$$

In the configuration, it's also possible to choose any overall consumption. In this case, the home consumption is only based on the default goal consumption given at the beginning.

2.3 Progress of a simulation

At the beginning, each smart house launches necessarily two agents: *home* agent and *controller* agent. The *home* agent declares itself to other houses already launched. An outside agent called *life conditions* allows to give the current simulation time, each home is based on the same time. The *controller* agent waits for new devices until the total number is respected and until the home time is synchronous with the current time simulation. For each new device, it declares itself to *controller* agent, and gives all its features on a directory. Both processes are shown in figures 1-2.

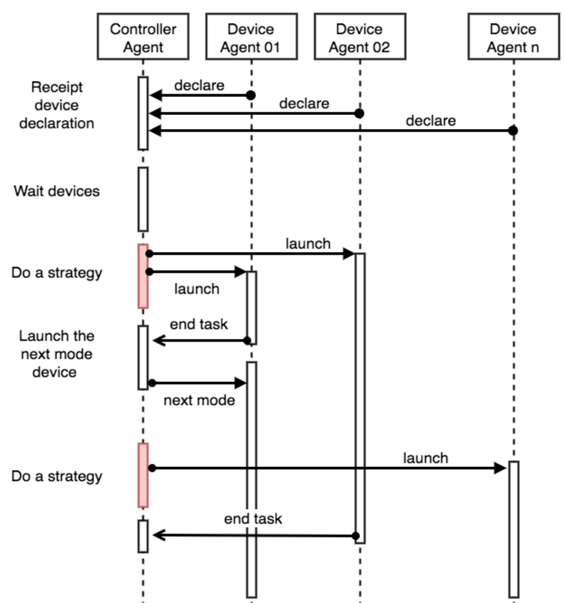

Fig. 1. Process of controller agents

Once the simulation is launched, the *controller* agent at a discrete frequency, computes the current best strategy according to the goal consumption and launches devices if necessary. Then, the launched devices give feedback at the controller to report its values.

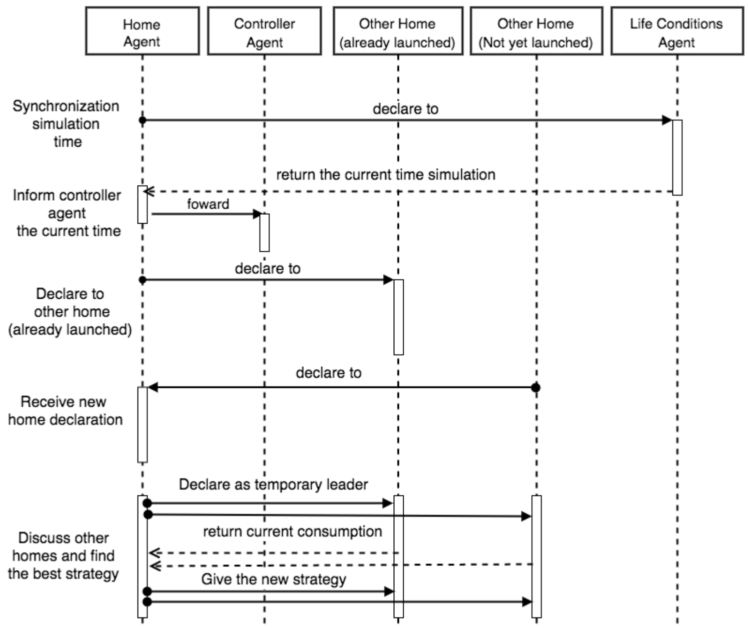

Fig. 2. Process of home agents

During the same time, the *home* agent discusses with other homes about the process of leader-follower. For voting the new current leader, a random timeout value is attributed at each home. The first home where the timeout is finished, declare itself to other home as leader, waits answers with the house consumptions and establishes the new global strategy. Then it sends the new goal consumption. The current leader lost it status.

Sometimes, the communication between houses is broken, to avoid to block the process, the leader starts a timeout, in the case where homes don't give answer, the leader computes their new strategy. It believes their consumption by an average of the last known house's consumptions.

At the end of a simulation, all consumption for each home are stored in an individual Comma-Separated Values file, where each line indicates the time in seconds and days, number of activated devices, the current goal consumption and the current consumption.

3 Results

Two cases are shown in this paper. Each home returns CSV file with it consumption, a Python script allows to join consumption, analyzes them and plots curves. For each consumption result, two curves have been plotted, one represents the real consumption, and the other shows the goal.

3.1 A microgrid

In this test, consumption's goal evolves in function of the hours of a day.

Simulations characteristics:

- 5 houses
- 15-25 devices per house
- goal consumption:
  - in $[\![10h, 18h]\!]$, quadratic function whose the max is 5000
  - else, 3000
- 5 simulated days

The regulation (8) is applied according to their current consumption. In the figure 3, the two curves are similar (the blue one for the goal, the red one for the consumption), which means that the neighborhood and home's consumption are regulated within a goal.

For a smart house, the average consumption have only 1.2% error compared to its goal. The minimum of the whole simulation have less than a 10% error and the maximum less than a 15% error. 90% of all values have less than 5% of error compared to each house goals. But it is important to evaluate and catch errors and limitations. The next simulation tests its limits.

3.2 Limits

All simulation get unwanted behaviors, in order to identify these ones a test has be done with the following characteristics:

- 5 homes
- 25-40 devices per home
- Goal consumption:
  - in $[\![10h, 18h]\!]$, quadratic function whose the max is 7500
  - else, 2500
- 7 simulated days

In the figure 4, the top chart represents global consumption of the neighborhood. Below two house consumption's curves are available. In both chart, the blue graph is the goal and the red graph is the consumption. At the top chart, a gap between the goal value and the system response happens when the consumption need to be regulate at the minimum goal.

The regulation isn't perfect, there are several reasons for possible error:

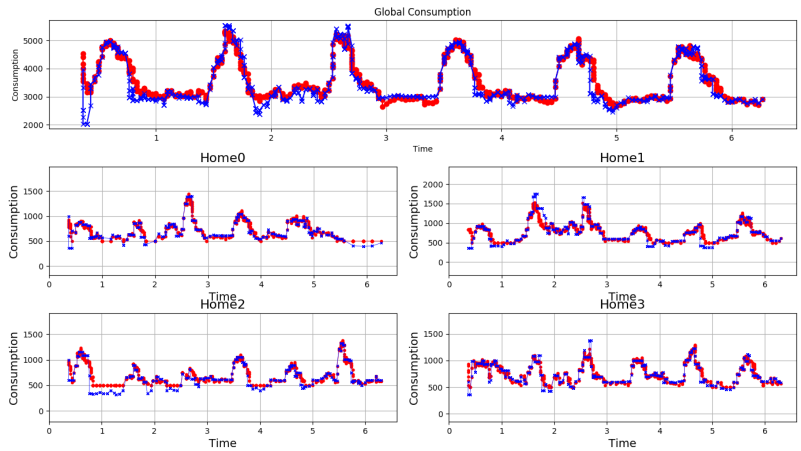

Fig. 3. Power production available in the neighborhood

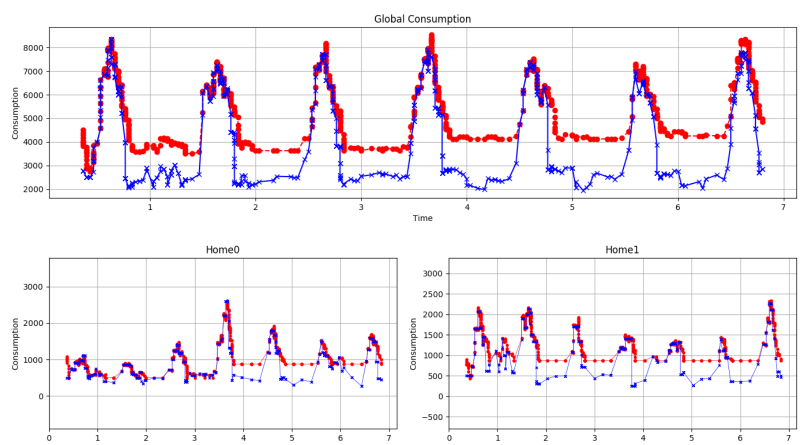

Fig. 4. Power production available in the neighborhood

– The dynamic programming knapsack problem resolution has to create a $n*m$ matrix, where n is the number of device and m equals at weight. To reduce the matrix size, each weight is rounded to the nearest tenth.
– For acyclic devices, consumption is determinate through a markov chain.

During this test, the system chooses to regulate the consumption of all homes to 500 but homes can't be regulate at this value. This notion of physical problem is really important to know the limits of the simulation and to highlight potential problem during the implementation of this system in the real life.

These results have strengthened confidence about the presented model in the architecture and process chooses to realize this simulation. These different results shows the possibility to easily create scenarios with various parameters, strategies and goal consumption.

3.3 Enhancement

The behaviors of users are limited, the user gives use's preferences for each device. To raise the realism and flexibility of this model, it is important to add users perturbations. So that the model will be developed on Gama Platform[3], facilitating the interface creation and interactions with the simulation. The final goal is to create a simulation more adaptable that can change easily before or during the test, and the possibility to interact with the simulation to disrupt the routine execution and verify that implementation can work in real life.

4 Conclusion

This paper has highlighted the necessity to implement some flexible demand-side management strategies. The simulation regulates the consumption on a local scale (house) and global scale (neighborhood). The results of this study indicate that the consumption can follow an order with the condition that the consumption is schedulable. In first though, a toy model is done. This one works in good conditions with few errors, few perturbations and few changes. The next step would be to add some human disruptions in order to raise the realism and the credibility.

Acknowledgment

One of the authors is in an engineering school (France, same degrees as Msc). He works in a half-time curriculum with an associated professor about his subject: a multi-agent model for microgrid applications. This paper concludes his curriculum.

[3] Official website: http://gama-platform.org

Bibliography

[1] Eremia, M., Toma, L., Sanduleac, M.: The smart city concept in the 21st century. Procedia Engineering **181**, 12–19 (2017)

[2] Ahat, M., Amor, S.B., Bui, M., Bui, A., Guérard, G., Petermann, C.: Smart grid and optimization. American Journal of Operations Research **3**(01), 196 (2013)

[3] Dimeas, A.L., Hatziargyriou, N.D.: Operation of a multiagent system for microgrid control. IEEE Transactions on Power systems **20**(3), 1447–1455 (2005)

[4] Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi-agent systems with jade. In: International Workshop on Agent Theories, Architectures, and Languages, pp. 89–103. Springer (2000)

[5] Kofinas, P., Vouros, G., Dounis, A.I.: Energy management in solar microgrid via reinforcement learning using fuzzy reward. Advances in Building Energy Research **12**(1), 97–115 (2018)

[6] Guérard, G., Pichon, B., Nehai, Z.: Demand-response: Let the devices take our decisions. In: SMARTGREENS, pp. 119–126 (2017)

[7] Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the knapsack and sum of subset problems. Journal of the ACM (JACM) **22**(4), 463–468 (1975)

[8] Arbaz, A., Waqas, M., Shehzad, K., Mahmood, A.: Home energy management and knapsack technique in smart grid environment. In: Proceedings of 2nd International Conference on Engineering & Emerging Technologies (ICEET), Lahore, Pakistan (2015)

[9] Khan, A., Javaid, N., Ahmad, A., Akbar, M., Khan, Z.A., Ilahi, M.: A priority-induced demand side management system to mitigate rebound peaks using multiple knapsack. Journal of Ambient Intelligence and Humanized Computing pp. 1–24 (2018)

[10] Rahim, S., Javaid, N., Ahmad, A., Khan, S.A., Khan, Z.A., Alrajeh, N., Qasim, U.: Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources. Energy and Buildings **129**, 452–470 (2016)

[11] Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE **95**(1), 215–233 (2007)