UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

AGENCY SERVICES FOR THE MANAGEMENT OF DISTRIBUTED ENERGY NETWORKS USING PARALLEL AUCTION MARKETS

by

Ignacio José López Rodríguez

A thesis submitted to the

University Institute of
Intelligent Systems and Numeric Applications in Engineering (SIANI)

University of Las Palmas de Gran Canaria

for the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

September of 2015

AGENCY SERVICES FOR THE MANAGEMENT OF DISTRIBUTED ENERGY NETWORKS USING PARALLEL AUCTION MARKETS

University Institute of
Intelligent Systems and Numeric Applications in Engineering (SIANI)

University of Las Palmas de Gran Canaria

A thesis presented by

Ignacio José López Rodríguez

Directed by

Dr. Francisco Mario Hernández Tejera

The PhD student | The director

Las Palmas de Gran Canaria, September of 2015

A mis padres y mis hermanos; a ese todo del que me siento parte hasta los huesos.

A mis tías, María Dolores y María Victoria.

Abstract

Facing growing energy demand, the exhaustion of fossil fuels, and the effect of greenhouse gases requires moving towards a new model of electrical grid. Aware of this situation, the US and Europe's governments are working on the development of the Smart Grid, which is defined as a distributed, reactive and intelligent grid that will allow modulating the demand and use of available power generation dynamically, thus facilitating the integration of renewable energy sources safely and efficiently. One of the most important aims of the Smart Grid is to increase the participation of users in the management system of the network, who, by using smart local devices, are envisioned to be able to configure their consumption habits according to energy prices and to contribute in the generating facet.

The enormous dependence of modern society on energy supply provokes the need for a gradual transition that converts the Smart Grid into a long-term process. The initial stage of this transition is supported by small distributed energy networks that are designed to be self-managed areas of the distribution network composed of modular loads and distributed energy sources. These networks represent controlled environments for the inclusion of renewable energy sources and the installation of distributed and reactive management systems such as those envisioned in the Smart Grid. The Chapters 1 and 2 are devoted to describing the electrical grid and the Smart Grid, including their present form, mission and challenges.

The characteristics of intelligent agents make them a particularly suitable technology for managing environments in a distributed, reactive, intelligent and participative manner. Accordingly, there are many voices advocating the implementation of the management system of the Smart Grid as a multi-agent system. However, though intelligent agents have been a promising technology for the last fifteen years, the truth is that, far from achieving this, the

technology has failed to establish itself as a practical solution in many technological settings that demand its functionality, as the Smart Grid now does. Among these are computational grids, P2P networks and virtual organizations, which have opted for more practical solutions, relegating the intelligent agents to the academic sphere.

Aware of this barrier to entry, this thesis designs and implements an architectonic model called *Agency Services* (Chapter 3). This aims to facilitate the easy and realistic integration of software agents into the virtual environments that have arisen as result of the recent advances in information and communication technology, being especially suitable for implementing electronic markets and management systems based on coordination and negotiation actions, such as those expected in the Smart Grid. The Agency Services model is heavily based on the Cloud Computing paradigm, which has proven to be the type of solution that users adopt in practice. Also, the new model takes important lessons from solutions that have succeed in similar environments, such as computational grids and P2P networks.

This thesis also provides a detailed review of the most important algorithms of the literature for the implementation of energy markets in distributed energy networks (Chapter 4). The review is particularly focused on agent-based solutions because they are the unique approach to empower users and implement reactive and distributed solutions. The review classifies and studies the literature according to the present needs of energy markets, highlighting the most complete proposals and discussing the subjects on which work remains to be done. Finally, an algorithm based on reverse parallel auctions has been selected as the most suitable for achieving distributed, flexible and reactive management systems based on autonomous entities.

The combination of both the architectonic solution and the energy market's algorithm is evaluated in a novel co-simulation infrastructure specially designed for this purpose, which combines the best solutions of both worlds, multi-agent systems and the electrical grid (Chapter 5). The experimental evaluation is based on demand-response programs, which are enriched with a new conceptual model based on critical loads and negative loads that opens the door to the implementation of market mechanisms (Chapter 6). Demand-response programs are chosen in particular because they are one of the most immediate and realistic milestones on the road to the Smart Grid. The virtues and benefits of the Agency Services model are proven both quantitatively and qualitatively. Specifically, the Agency Services model has been shown to facilitate the installation of distributed agent-based solutions in restricted environments, to simplify the technical requirements of the client's facility, and to improve the reliability of the system. Furthermore, in the particular case of energy management, it has been proven that energy balancing is successfully accomplished through negotiations between software agents.

As the experimental evaluation reveals, the effectiveness of parallel auctions falls dramatically as the distribution of buyers between auctions becomes less uniform. This fact, which has not been addressed in the literature so far, may turn parallel auctions into a useless management system. To solve this problem, this thesis designs and implements a novel solution that manages to distribute buyers properly (Chapter 7). The mechanism noticeably improves the effectiveness of parallel auctions and is guided by a set of rules that preserves market competence. Furthermore, it is tailored to the needs of large and highly distributed environments, such as the Smart Grid and computational grids.

The solution presented in this document is fully compliant with the main energy standards. In particular, the Agency Services model is compliant with the *Energy Interoperation* model defined by OASIS. As for DR programs, they are implemented using the OpenADR standard, which is the most widely adopted solution by sellers in this matter. Furthermore, it is noteworthy that all work developed in the course of this thesis respect the principles of the *reproducible research* movement, so that it can be evaluated and verified by other searchers.

Contents

List of Figures

List of Tables

¡Estoy hasta los huesos de tu cuerpo!. . . ¡De tu carne de hombre que no aguanta los tiempos!. . . ¡Ni aguanta el sol de estío!. . . ¡Ni los fríos de diciembre!. . . ¡Para esto crié yo mis pechos, duros como el pedernal!. . . ¡Para esto crié yo mi boca, fresca como la pavía!. . . ¡Para esto te di yo dos hijos, que ni el andar de la caballería ni el mal aire en la noche supieron aguantar!

"*La familia de Pascual Duarte*", Camilo José Cela.

Su nombre amontona pasado en mis ojos.

"*Ester Primavera*", Roberto Arlt.

CHAPTER

1

The electrical grid

The prolific inventor Thomas Edison introduced the first electric power system in New York City in 1882. The *Edison Illuminating Company* operated with direct current at 110 volts and initially supplied light to 59 customers in the Wall Street area. Unaware of the important social benefits that would arise later, the main goal of Thomas Edison was to create a new profitable business. By the end of 1880s, Edison had sold the patent for generating and transmitting electricity, and many cities of US and Europe had set up many similar small central stations capable of supplying few city blocks. However, due to the use of direct current, the range of these first generating stations was limited to a couple of kilometers. In 1888, Nikola Tesla, a former employee of Edison, received a patent for the induction motor, which would enable the high-voltage transmission of alternating current over long distances with low losses. In 1896, Nikola Tesla, by then working for *Westinghouse Electric & Manufacturing Co*, turned this idea into reality by constructing a hydroelectric station in the Niagara Falls that was capable of transmitting significant amounts of power to Buffalo, New York, more than 32 kilometers away. In the end, this innovation would establish the foundation of an electrical grid designed for centralized generation and distributed loads, a concept that has endured to the present day.

The need to meet users' demand, whatever its amount, and to do it immediately are also essential features of the electrical grid in its present form. In particular, this conception implies that each time a user turns on a device, such as a light bulb, the grid must be prepared to supply the electrical energy that it requires. No doubt, these two characteristics, high availability and reactivity of energy supply, have been decisive to the extraordinary development of modern societies. However, in the long term, the application of this concept to a resource such as the electrical energy, which is expensive, scarce, and very difficult to store on a large scale, carries many problems and faces a future of uncertainty.

This chapter presents a brief introduction that aims to cover all these issues. It reviews the fundamental concepts of the electrical grid, including its infrastructure, the crucial mission it fulfills, the market forms it gives rise to, and the important challenges it must confront in the near future.

1.1 Mission

In the period known as *Second Industrial Revolution* in the late 19th century, modern societies became increasingly technical. This process was strongly influenced by the new concept of social welfare, which among its pillars included and effective public health system, quality of employment and comfort. Most advances in these areas over the last century use electricity as motive power, assuming the existence of an electricity source that is continuous, instantaneous and reliable. As a result, the dependence of modern societies on electricity has grown to the extent that, today, its use is correlated with the gross national product: countries having difficulties to adequately supply all facets of society with electrical energy (henceforth also referred to as *energy*), show much higher levels of poverty. According to the projections of the *International Energy Agency* (IEA), nearly 90% of global energy demand growth from now to 2035 will come from non-OECD countries [IEA14]. That is, the progress of developing countries is related to the parallel development of the infrastructures needed to generate and supply energy reliably.

Over time, the dependence of modern societies on electricity has continued to grow, thus giving rise to a dangerous dependency loop: modern economies need more electricity to progress, which in practice means achieving new advances entirely dependent on the electricity supply. All the weight of power supply rests on a large and complex infrastructure known as the *electrical grid* (also referred to as *power grid*). Formally, its mission is to meet the electrical energy demand of all customers in a reliable and secure manner, regardless of time or place. If a customer switches on an electrical device such as a light bulb, the responsibility of the electrical grid is to generate, transmit and deliver the energy needed to power the bulb at that very moment, and do so transparently to the user. As will be shown, several physical and technological factors make this task complex and difficult to implement efficiently.

1.2 Structure

One of the most outstanding characteristics of the existing grid is that it works in a highly centralized manner: the electricity is generated in a few large power plants that are connected to the customers through transmission lines. The whole process is usually described in three stages:

1. *Generation*: Energy is generated in power plants that are far from the urban centers and, when necessary, close to places rich in natural resources such as coal or wind.
2. *Transmission*: The electricity voltage is raised to very high levels. Then the energy is transported from the power plant to the proximity of urban centers and industrial areas.
3. *Distribution*: The electricity voltage is significantly reduced. Then the energy is delivered to the bulk of consumers, which are concentrated in the domestic and commercial sectors.

The transmission and distribution phases fulfill different objectives, so that each requires its own type of network:

- *Transmission network*: Covers long distances minimizing the losses arising from energy transport. To this end, the transmission network works with very high voltage (HV) levels, and its infrastructure is built of overhead lines made of copper or aluminum, which present low resistance.

- *Distribution network*: Performs the final delivery of electricity to the customers. The distribution network is composed of medium voltage (MV) and low voltage (LV) lines that, ultimately, inject the electrical power at the voltage level required by households and commercial and industrial customers. Although distribution lines are mostly overhead, in recent years the installation of underground lines is gaining popularity in order to avoid exposure to climate events and other risky incidents.

The electricity can be transported and delivered effectively thanks to the alteration of the voltage. This task is performed at specialized facilities called *substations*, which are located throughout the grid. In particular, voltage transformation is necessary in each of the delivery stages:

- *Generation*: Energy produced by generators is usually at tens of thousands of kilo-volts (kV). Prior to be injected into the transmission network, the voltage is increased to hundreds of thousands of kilo-volts in a transmission substation located at the power station. The initial voltage of the transmission network depends on the distance and the amount of electricity to transport (800 kV being the maximum that can be achieved).
- *Transmission*: The voltage level is successively reduced in transmission substations until reaching the boundary defined by the distribution network, which is usually set at 115 kV or 132 kV. The step down of voltage depends on the distance and the need of branching sub-transmission networks.
- *Distribution*: The voltage is significantly reduced in a distribution station to the levels required by the medium and low voltage networks. As in the transmission stage, the voltage can be reduced more than once. Before the energy is passed to the meter attached to each household, the voltage is decreased to 240 volts or 230 volts by a small equipment called *transformer drum*.

Besides operating on voltage, substations also have other functions in the grid, such as multiplexing distribution in various directions and, if necessary, disconnecting individual lines or even complete zones from the transmission network. Figure 1.1 depicts the structure of the electrical grid described so far.

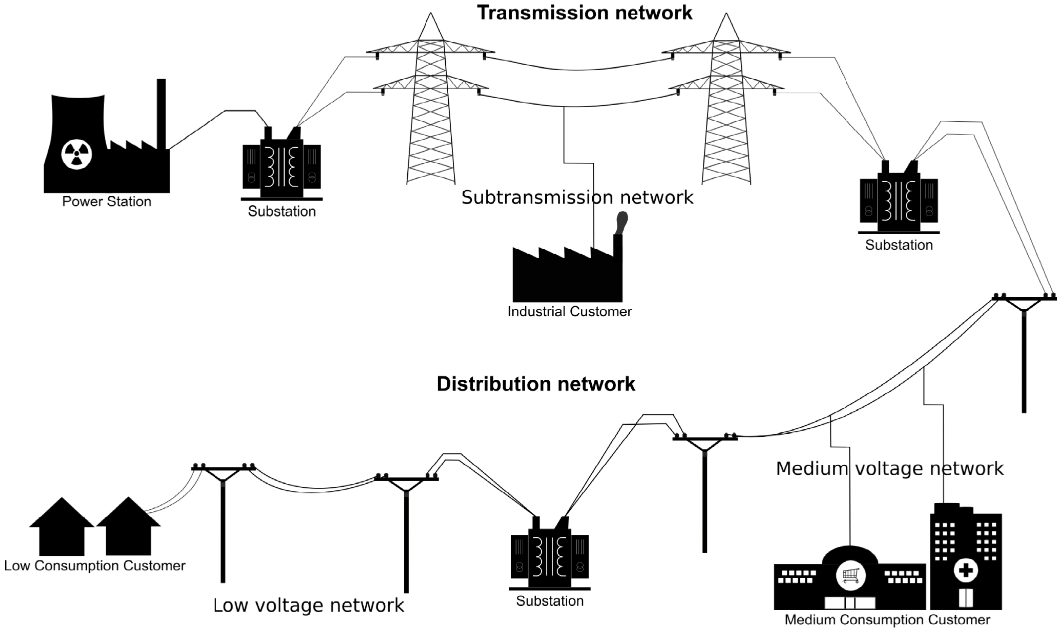

Figure 1.1: Overview of the structure of the electrical grid.

1.3 Why such a centralized model

The first electric power system introduced by Thomas Edison, in 1882, somehow worked as a decentralized system in which small scattered supply systems were in charge of feeding the lighting system of a few city blocks. This period is formally known as the *first revolution* of the electricity sector. In 1888, when Nikola Tesla introduced the alternating current, and thus the high-voltage transmission, it opened the door to switching to a centralized model of generation. In fact, in the long run this innovation would change both the existing grid's structure and the way of marketing the energy, and would mean the beginning of the period known as the *second revolution* of the sector.

The ability to transmit energy over long distances led power stations to move nearer to mines and waterfalls, where large amounts of energy could be generated at lower costs. That is, companies of the sector took advantage of the ***economies of scale*** principle. As society was profiting from the new generation capacity, new business, services and activities grew around it, thus beginning to take shape today's modern lifestyle. From the 1920s, the power supply became an important factor in national economies, so government's prevailing view was to make electricity as available as possible. This trend also created a dependency loop: developed economies demanded more electrical energy to keep growing around the electrical grid.

After the 1930s, with the aim of lowering costs and securing the reliability of supply, US and major European countries (e.g., France, Germany and UK) strengthened the regulation of the sector and significantly increased their participation in building power plants, transmission networks and backup systems. Gradually they adopted the idea that the generation, transmission and distribution of electricity should be considered a natural monopoly, so that in most of the countries small generation companies were pushed out of the sector or grouped in large utilities to which governments entirely delegated the task of supply. Thus, the concept of electricity as a public service provided by utility companies was consolidated.

Accordingly, the factors that drive the electrical grid to a highly centralized structure were chiefly of economic nature [SJ06]:

- Reducing generation and transmission costs by concentrating the production in a few large power stations (principle of economies of scale).
- Supporting national economies by generating large amounts of energy with power stations located near mining sites and hydro-power sites.
- Saving on investing in backup units by integrating all power stations and transmission lines into a single system.
- Facilitating the management tasks, and thus the reliability and safety of the grid.

1.4 Operation

The figure responsible for providing a reliable and quality grid is the *System Operator* (SO). Since the liberalization of electricity sectors, in which the transmission and distribution of energy remain a natural monopoly, there is a broad consensus that the SO should be implemented by a non-commercial, neutral and independent entity. In this specific case, the operator is referred to as *Independent System Operator* (ISO). The responsibilities assigned to the ISO include, importantly, ensuring the impartial access and use of the network by the players involved in the generation, distribution and market of the electrical energy. On the other hand, the management of the transmission and distribution networks is assigned to different roles: the *Transmission System Operator* (TSO) and the *Distribution System Operator* (DSO).

Many of the functionalities of the ISO are provided in form of ancillary services to all grid players. The most important of these services, and ultimately the main aim of the operation of the electrical grid, is to **match supply and demand dynamically**, so that both are continuously balanced. This objective requires intensive supervision and the presence of complementary services, such as scheduling of available resources in different time scales, contingency plans, managing reserves, and voltage and frequency control.

As mentioned before, the health of the electrical grid and most of its components mainly depend on the correct balance of supply and demand at all times: the amount of power injected into the grid must be continuously equal to the amount of power consumed by all loads. Otherwise, either the surplus or deficit of production may destabilize the system and jeopardize equipment, thus resulting in local blackouts or cascading blackouts. Achieving this goal is especially difficult because demand is dictated dynamically by clients. In addition, the demand is made in a way that cannot be predicted perfectly, so the balance has to be achieved through the use of a load-following control scheme; that is, supply must follow load. Energy balancing is performed in real-time, 24 hours a day and 365 days a year. Furthermore, the fact that energy cannot be stored on large scale makes the work more difficult, because in practice the production must be consumed or discarded. In practice, all this complexity is dealt with through a hierarchical control scheme designed to achieve the balance of energy in different scale levels and in different time frames. Normal grid operation distinguishes three levels that are formally known as: primary, secondary and tertiary control [1] [NER11].

The **primary control** is the first-line corrective action in disturbances. It is intended to restore small deviations between power generation and consumption that can be tackled in a few seconds. The key measure to identify small imbalances is the generator's output frequency, as disturbances deviate frequency from that

[1]There is no consensus in the literature about the number of levels. Many works do not differentiate the second and tertiary levels.

established by the system standard, which is 50 Hz in Europe and 60 Hz in the US. In more detail, if production exceeds demand, the rotational movement of the generator speeds up, thus increasing the frequency. Conversely, if there is a temporary production deficiency, the generator's rotational speed slows down and frequency decreases. It is important to note that the primary control does not return frequency to normal, but only stabilizes it.

When the energy imbalance cannot be solved by small corrections of individual engines, it is necessary to put into action a coordinated plan that involves generators from multiple power plants. This kind of action belongs to the realm of the **secondary control** level. It usually takes few minutes and aims to restore the minute-to-minute balance by setting the frequency to the standard value of the grid after it has been normalized by the primary control.

To facilitate the control mission, some national grids are divided into control areas. Each area is governed by the figure of *Balancing Authority* (BA), which is in charge of ensuring the operation within an area through the development of resource plans, maintaining the balance between load and generation, controlling transmission flows and voltages, and ensuring that frequency is held within normal limits. When balancing areas cannot meet the demand using their own resources, or there is surplus of production that can be consumed in other regions, BA can manage exchanges with neighboring control areas. The number of BAs, as well as the regions they run, is determined by the system operator. In the US there are over 130 balancing authorities, although over the last past several decades, motivated by economies of scale, they are gradually getting larger as a result of the union of some of them. The common rule in Europe is that each national grid works as a unique control area that can exchange energy with connected neighboring countries. On the other hand, correcting the deviations of balancing areas is based on the value *Area Control Error* (ACE), estimated in MW. ACE is a measure of error in the system frequency that helps to identify differences between the actual and the scheduled net power flow within a control area. A positive value of ACE means that generation within the area exceeds the load by more than the expected value. In this case, the generation within the control area must be reduced. Conversely, negative ACE means local generation must be increased.

The elimination of ACE requires the coordinated action of multiple generators of multiple power plants. These kind of actions can only be run with an overview of the control area. This information gap is covered with an advanced system called *Supervisory Control and Data Acquisition* (SCADA). Making use of devices and sensors deployed across the network, SCADA continuously collects information on the grid status. In particular, with a typical periodicity of four seconds, it collects data on the system frequency, the generators and net real interchange between the system and adjacent systems. Using this information, the *Automatic Generation Control* (AGC) subsystem is responsible for setting ACE next to zero. The AGC software is run on the area control center to determine the most reliable, stable and economical solution, which basically consists of the resources that must participate in the restoration, as well as the set points to which devices must operate. Generators equipped with AGC devices are informed of the new set points through the communication network, after which they proceed to readjust their configuration. To achieve a reactive control system, generators that participate in AGC actions generally have fast response times and flexible production levels. In case the capacity of the AGC subsystem is not sufficient to cover the imbalance, the direct action of human operators may be necessary so that the system operator can phone the generation operators and ask for a change in the output of the power station.

When resource planning or demand estimates fail so much that actions of the secondary control are not sufficient to restore the stability to the area, the **tertiary control** comes into play. It is intended primarily to address contingencies that require solutions that last from 15 minutes to several hours. Common actions of the tertiary control are enabling reserves, rescheduling net interchanges of the control area, and shedding parts of the load when necessary. This control level is not defined in the same way around the world, so sometimes this is understood to be part of the secondary control. Tertiary control can certainly be seen as a sort of long-term secondary control, but it is clearly differentiated by the type of resources it uses and the duration of its action.

So, as depicted in Figure 1.2, energy imbalances are addressed gradually in three levels, which are mainly characterized by the period in which they run, and the size of the imbalance they can solve: from the first level, which is targeted to small imbalances that can be corrected quickly, to the third level, which handles large imbalances that may take hours to be corrected.

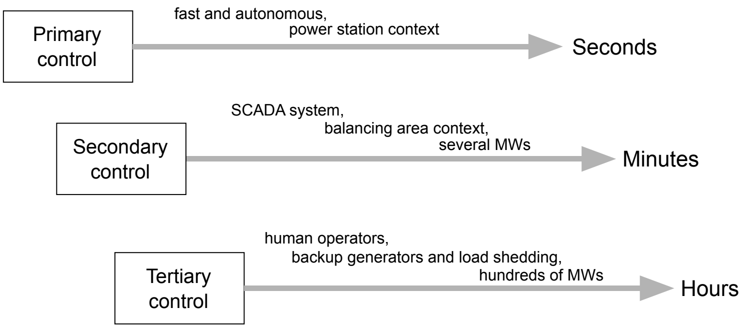

Figure 1.2: General properties of the primary, secondary and tertiary controls.

Once the market or the system operator has assigned generating plans to power stations, the latter are responsible for determining how their generating units must meet the demand at all times of the day. That is, the power station has to set which units are going to be used, at which periods of the day they must operate, and which must be their production level at each moment. This task is always approached as short-term optimization algorithms that, beyond the essential purpose of supplying the required power, are focused to minimize overall fuel costs while satisfying the constraints imposed by the system. Specifically, short-term scheduling is tackled through two processes:

- *Unit commitment (UC)*: This determines the time points at which each generating unit must be started up and shut down, as well as the amount of energy it should produce when is on-line. Unit commitment is typically done one day ahead.
- *Economic dispatch*: This determines the set-points of each of the on-line generating units in order to meet the existent load at minimum cost. The economic dispatch optimization algorithm is typically run every 5 or 10 minutes.

The efficiency of these processes is crucial for large utilities: a reduction of less than 1% can result in savings of millions of dollars a year.

1.5 Demand-response programs

Rather than increasing generation capacity, decreasing demand is the most desirable measure when balancing energy. As described in Section 1.7, a significant amount of the investment on the generation, transmission and distribution of energy is driven by the need to cover the demand in high peak times, which in practice means less than 1% of the demand during the year. Also, avoiding using generation capacity, especially that installed to cover demand peaks, implies considerable savings on energy sources that are expensive and inefficient by nature. Aware of these conditions, the electrical grid has developed mechanisms to combat rise in demand. They are mainly two groups of means:

- *Demand-response (DR)*: This is an action by which end-nodes are encouraged to make short-term reductions in response to price signals, or as a result of bilateral contracts. After receiving a DR signal, typical actions are turning off banks of lighting, adjusting HVAC levels or shedding part of the demand of industrial processes.
- *Demand side management (DSM)*: This involves measures intended to improve energy efficiency, which are mainly related to the functioning of consumer devices.

DR programs are one of the most promising mechanisms for efficient energy balance. However, due to the lack of modernization of the electrical grid, its application is still rather limited: it is mainly focused on discarding load from factories and large facilities under contracts previously agreed with the system operator. In particular, the facilities involved, in exchange for a payment, access to discard part of their demand under certain circumstances and in specific time periods. This type of solution lacks flexibility.

With the aim of bringing DR programs to all customers, and thus obtaining all the benefits they can really offer, the community has long been working on DR standards and devices. The OpenADR standard [OAD] is the most advanced proposal, for which most popular sellers are already offering products.

1.6 Liberalization and energy markets

Since the beginning of the large-scale generation of electrical energy in the late nineteenth century, up until less than two decades ago, the power supply has always been treated as a natural monopoly: as a service that should be provided by the government through utilities. The electricity sector, due to the complexity of both its infrastructure and operation, has always been understood as a field in which it is not easy to successfully apply market principles. In addition, the enormous importance that electricity has acquired for economic and social development of countries led governments to take on its management. However, this view of the electrical grid, partly because competition is limited or null, brings important drawbacks such as high costs, lack of innovation, and inefficiency. In contrast, the theoretical benefits commonly associated with the opening of electricity markets are: lower prices and operating costs, improvement of the system efficiency and service quality, fostering innovation, encouragement of the use of clean energy solutions, and increment of the array of energy products available to consumers. With these goals in mind, most developed countries have begun liberalizing the electricity sector, as well as creating and opening electricity markets, which is a complex task due to the magnitude of the system, long-established traditions, and the need to continuously provide a reliable service.

In particular, the liberalization of the sector is focused on the definition and implementation of a framework that separates the activities that can successfully operate in competition and those which, by nature, must remain as natural monopolies. In general, the guidelines that the emerging liberalization process follows are:

a) Deregulating generation and supply activities, and allowing the entry of private agents in these stages. The idea is that generators sell energy in a wholesale market, and both retail companies and large industrial customers buy it in order to offer supply to customers in the domestic and commercial sectors at regulated prices.

b) Transmission and distribution networks remain as a natural monopoly. The management of the network is delegated to an independent operator that must guarantee its impartial and non-discriminatory use for generators and retailers.

This scheme aims at breaking the strong vertical integration typical of the state-owned electrical grids, in which a single company usually takes presence in all stages of the energy delivery: generation, transmission, distribution and supply. By contrast, it is replaced by a model that pursues to introduce competition in the generation and supply activities. Accordingly, the sector restructuring is formally guided by two key lines:

i. *Vertical unbundling*: Avoiding the simultaneous presence of agents in multiple stages of the generation and delivery process. That is, to ensure and reinforce competition by avoiding that a player can be a customer of itself at a subsequent stage. Actually, if it is not properly regulated, an agent, in the role of retailer, may buy the same energy that it produces as generator, thus possibly distorting prices.

ii. *Horizontal separation*: Reduce the horizontal concentration and guarantee the offer in the generation and supply activities by stimulating the entry of agents at each level.

This model is intended to create **competition** through wholesale and retail markets. In the former case, retailers and large industrial consumers buy electricity directly from generators; while in the latter case, retailers wrap electricity from the wholesale market in commercial products that re-sell to end-use customers. Due to the critical nature of electricity to society, tariffs at this latter stage are regulated.

The main trading mechanism in retail markets are long-term contracts between retailers and end-use customers. Specifically, the customer contracts an electricity product that, according to the profile of the demand, guarantees a fixed price for the energy. In this way, end-use customers are protected from the high volatility of energy prices. On the other hand, wholesale markets are based on rules and mechanisms typical of commodity markets. In this case, the fact that electricity cannot be stored on a large scale along with the impossibility of providing it at the instant customers demand it, makes it necessary to work with estimates and hold markets in advance. Actually, this type of market involves most of the wholesale activity. There, a central authority initiates a market in which generators and retailers, according to demand estimates, negotiate a specific volume of electricity for specific future blocks of time. According to the duration of these periods of time, three types of spot markets are held:

- *Day-ahead*: This schedules the production and consumption for the next day. This market is organized on an hourly basis: players submit offers for purchasing and selling energy for blocks of hours of the day-ahead. Generators make offers for the hours they have available capacity, while retailers submit bids for purchasing energy in accordance to the estimated demand. Market players commonly put in 24 bids per day.
- *Hour-ahead*: This adjusts the deviations on the supply and consumption with regard to that initially scheduled in the day-ahead market. Several market sessions are programmed per day, which are intended to correct the deviations for specific time periods. The players are usually the same as in the day-ahead market. This market is also known as *Intra-day market* or *Adjustment market*.
- *Real-time*: This acts as the last economic level for achieving the balance between supply and demand. It is not based on the same market mechanisms that the day- and hour-ahead markets use. Instead, in the real-time market, generators and consumers submit bids that specify the prices they require to vary their supply or demand for a specific volume in a short period of time. This market is also known as *Balancing market*.

At the closing time of day- and hour-ahead markets, after collecting all offers and bids, a central authority proceeds to clear the price for each block of time. Clearing algorithms of electricity markets are mainly based on the marginal cost of generation. In terms of trading volume, the day-ahead market is the principal mechanism for scheduling the energy dispatch of each day. Next, hour-ahead and real-time markets are used respectively to correct deviations from the initial plan and to balance the supply and demand minute to minute. Furthermore, countries that have made most progress in the liberalization process are increasingly integrating markets of ancillary services. The aim of these is to trade capacity and functions that system operators use to cover unplanned imbalances and incidents.

With the onset of electricity markets, has also emerged the figure of *Market Operator* (MO), which is the entity responsible for the management of the markets. The MO is commonly related to the ISO, and is complemented with the *Energy Regulator* (ER). This second role is established by the government and its principal mission is to ensure that market operation occurs in compliance with the government's regulation, paying special attention to the parts of the sector that remain natural monopolies, such as the transmission and distribution networks. In most cases, the MO is in charge of the management of the day-ahead and intra-day markets, while the balancing (real-time) and ancillary services markets are delegated to the ER.

1.7 On road to obsolescence

The existing grid shows clear signs of obsolescence: the technological basis, the means of control, and the infrastructures for generating and delivering electricity have remained unchanged for decades. This lack of innovation contrasts with the context, which conversely has become particularly volatile. The huge increase in demand, the continuous fluctuation of prices and the inevitable depletion of fossil fuels are all challenges that electrical grids around the world have to face in the medium term. According to estimates, the coming decades will be crucial to overcome a threatening horizon [IEA14, EU14, EU13]:

- *Exponential increase in demand.* Between 2008 and 2035, world marketed energy consumption will increase by 53%. Although much of this growth is associated with emerging economies, developed countries will also experience a high rise in energy consumption. In particular, in Europe energy demand will increase by 60% from now to the year 2030.
- *Depletion of fossil fuels.* Most of the supply to meet the new demand will be based on fossil fuels. As a result, it is estimated that most of the conventional oil reserves will be depleted by 2030.
- *Lack of price control.* Most countries need to import fossil fuels. In addition, much of the fossil fuel resources are controlled by a small group of producing countries, which are mainly represented by the *Organization of the Petroleum Exporting Countries* (OPEC) and the *Gas Exporting Countries Forum* (GECF). The lack of local sources and the fact that main existing producers are living under unstable governments leads to lack of control of prices. The European case is paradigmatic: if the continent is not able to increase its energy production, 70% of demand will have to be met with external energy

sources over the next 20 years. As for the US, it will have to increase the production of crude oil by 13% by 2019 in order to combat the rise of prices.

- *Global warming*: Climate change is a fact. The *Intergovernmental Panel on Climate Change* (IPCC) has evidence that greenhouse gases have so far caused a global temperature rise of $0'6$ degrees centigrade. Moreover, the IPCC estimates that in the event of continuing abuse of fossil fuels, this temperature will increase between $1'4$ and $5'8$ degrees during the twenty-first century.

Also, characteristic factors of the grid, such as the low efficiency of generating mechanisms, or losses due to the transmission of energy over long distances, must be revised and corrected. For example, in relation to the amount of fuel used, the existing generating units only use between 25% and 40% of the energy generated. The rest of the energy is dissipated in form of heat during the process, which is difficult to transport. The efficiency ratio can be improved if generating units are near the place of consumption, so that heat rejection is applied right there. In this case, the efficiency ratio may reach 70%. As for the losses due to transmission and distribution of energy, the EIA estimates that they reach 7% in the US and 6% in the EU.

The August 14, 2003, the northeastern US and southern Canada suffered one of the worst blackouts in history. It involved some 50 million customers, and its economic cost was estimated at between 7 and 10 billion dollars. The causes were the combination of overloading and failures in the control algorithm. Far from being a single event, blackouts with significant consequences occur every year. The rise in demand and having to control an increasingly complex system make failures inevitable. However, the lack of reactivity of the grid and its monolithic structure causes small blackouts to become rolling blackouts, thus extending consequences throughout the network.

The most common cause of outages are temporary peaks of demand, which are usually focused on a small set of summer and winter days, when as a result of unusual temperatures there is a massive use of air conditioning or heating systems. These short periods of time may have a frequency of once every five years or even once every ten years, which makes reserve capacity extremely expensive. At the present time, between 25% and 50% of the electricity bills of most countries goes to finance the infrastructure in charge of covering usual peaks, which can mean activity that occurs during less than 1 percent of year. Moreover, in the current circumstances, this scenario can only get worse: according to the IEA, peak load will increase until 2050 by 28% in the OECD countries of EU, 15% in OECD countries of US, and 200% in China. This may be alleviated if it is possible to shed dynamically specific parts of the load during especially demanding time periods. However, in practice this option is usually only arranged with industrial and large commercial customers. Therefore, the concept of reliability of the electrical grid is highly inefficient, as it generally assumes responsibility to supply every load, regardless of its type and importance. The exceptions are the zones with essential public services, such as hospitals and police stations, which are especially protected so that they are usually the last to be affected.

Accordingly, the existing electrical grid is becoming obsolete:

- The Infrastructure, the technological basis and the control scheme of the grid have stayed the same for the past 60 years, with the result that they are not prepared to deal with the activity and complexity that the expected demand will entail.
- Generating units are highly inefficient, and the transmission of energy implies significant losses.
- The control system is highly centralized, so it losses reactivity and ability to act as it grows.
- The reliability and quality of service of the grid are virtually based on the concept of *all or nothing*. The need to supply outstanding peaks of demand charges the grid with excessive costs.

In view of the facts and forecasts, the community agrees that the existing electrical grid is not ready to meet the upcoming challenges, so that the entire conception of it, from the structure to the technologies used, must be revised.

De pronto, mientras contemplábamos la Ermita, se iluminaron sus ventanas y de la azotea surgió una llama que se elevó hasta el borde de los acantilados. [. . .] Ante nuestros ojos, la cosecha de muchos años de trabajo era presa de los elementos, y, al tiempo que la casa, nuestra obra volvía al polvo [. . .] y, sin embargo, en el resplandor de la llama había algo de alegría. Llenos de nuevas fuerzas, avanzamos de nuevo por el sendero. Todavía era oscuro, pero el frescor del alba ya ascendía desde los viñedos y los pastos. Y a nuestro corazón le pareció que los fuegos del firmamento amenguaban algo su siniestra violencia, pues en ellos se fundía la aurora.

"*Los acantilados de mármol*", Ernst Jünger.

CHAPTER 2

Distributed Energy Networks

The integration of new energy sources, the more efficient use of them, and expanding DR programs to households require moving to a new model of electrical grid. Aware of this, the US and Europe's governments are working on the development of the so-called *Smart Grid*, which is defined as a distributed, intelligent and reactive grid that will make it possible to modulate demand and exploit available power generation dynamically, thus facilitating the integration of renewable energy sources safely and efficiently. However, the fact that the electrical grid has not changed substantially over the past 60 years, along with the excessive dependence of developed countries on the energy supply, suggests undertaking a gradual transition that protects the security and reliability of the system.

With the aim of smoothing this transformation process, many efforts are being devoted to the creation of autonomous local areas in the grid. These areas, also known as *Distributed Energy Networks* (DEN), are conceived as a self-controlled enclosed cell of the distribution network composed of distributed generating sources, dispatchable loads and storage systems. DENs make it possible that distributed generation can co-exist with traditional infrastructures and standard control systems. This condition is especially important because:

i. It will be useful in facing the development of distributed energy environments gradually and within a delimited area.

ii. It reduces the need of making big investments with uncertain chances of profitability.

DENs must meet technological, economic, environmental and legislative requirements. It is therefore necessary to complement local control devices with an advanced management system that optimizes the operation of the DEN by planning, coordinating and monitoring the activity of the local generating units and demand devices. Accordingly, DENs are usually characterized by smart control systems that are able to balance energy dynamically. The aim of this transitional model is to build a grid composed of many linked self-supplied cells capable of exchanging energy between themselves. Thus, it is expected to satisfy large parts of energy demand in future.

This chapter reviews the main efforts devoted to the development and implementation of DENs. This review includes the management system, being particularly focused on its most ambitious approach, which proposes the use of distributed mechanisms based on market principles and intelligent agents. This chapter also discusses important challenges that this approach faces, which, unfortunately, have received scant attention from the research community. Finally, aiming to draw on comparative experiences, this chapter examines technological fields whose problems and challenges are similar to those faced by the Smart Grid. As a result of this evaluation, consolidated solutions, best practices and risks to consider are pointed out.

2.1 The Smart Grid

The symptoms of obsolescence pointed out in Section 1.7 can hardly be overcome by simply scaling up the current system. According to the European Commission, the solution to help surmount the energy crisis must meet three strategic objectives:

- *Sustainability*: This is necessary to adopt measures that help limit energy consumption in Europe, and boost the development of clean energy sources which will contribute to curbing the climate change.
- *Competitiveness*: The creation of competitive markets contributes to lower prices, thus benefiting the end-use customers. Moreover, transparent and well-

regulated markets attract investment and stimulate the search for more efficient solutions.

- *Supply security*: To ensure the reliability and security of supply, it is necessary to install mechanisms that help reduce demand and dependence on external energy sources. To this end, in turn, it is necessary to develop indigenous energy sources, and conceive mechanisms that make demand a more elastic good.

There is no single solution to solve all problems and challenges. Any solution designed to transform the future of electricity supply should consider a wide range of initiatives and technologies. Therefore, it is important to highlight that educating citizens on responsible consumption, and improving energy efficiency of appliances is as important as finding new energy sources.

One of the main lines of action to deal with the depletion of fossil fuels is promoting ***Renewable Energy Sources*** (RES), which essentially consist of wind power, hydro-power, solar energy, biomass and biofuels. RES are especially important in regions highly dependent on external energy sources because they are indigenous, clean and sustainable energy sources, offering competency to national energy markets. In fact, the two most cited benefits of RES for the electrical grid are: (*i*) enriching production mix and thus improving supply security; and (*ii*) reducing CO_2 emissions.

However, the current infrastructure and control system of the electrical grid are serious obstacles for the massive installation of renewable sources, as their integration entails important risks for security and reliability of the network. In particular, RES are considered non-manageable sources because:

i. The capacity to modulate their production dynamically is limited.
ii. They react abruptly to the voltage perturbations and the so-called *power dips*.
iii. They are stochastic so that the behavior of most of renewable sources is variable, with pronounced gradients.

As a result, in order not to jeopardize the grid, extremely conservative strategies are being used, with the result that the final share of renewable sources that can participate in the supply system is limited. Thus, in order to take full advantage of these sources, **more reactive control systems** are required: systems capable of dynamically adapting supply plans to the sources' availability.

Another important feature of RES is that they are not limited to use in large facilities. It is not unusual for warehouses, hotels, office buildings and even households to have small generating units, such as photovoltaic panels or small wind turbines, installed for self-supply. However, when these are connected to the grid, even though the owner may be rewarded by the operator, the energy is not commonly used because it has not been scheduled in advance. To insure that this energy can be real part of the supply chain, it will be necessary to have not only a more reactive control scheme, but also a **more decentralized** one: a control system distributed across multiple nodes in constant communication. This type of control would enable *Distributed Energy Resources* (DER) to contribute to the supply of the grid, thus including the activity of co-generation units, batteries and small generators. The contribution of this type of equipment would also help to improve the efficiency of the system, and curb the transmission and distribution losses.

Also, it is necessary to build a **more participative** electrical grid, so that users can actively contribute to balancing supply and demand. A priority in this regard is expanding DR programs to ordinary customers, since, as pointed out in Section 1.5, at the moment these type of programs are largely focused on factories and other specialized facilities that may produce high levels of demand. The central idea to changing this is that customers, based on pricing signals, can modulate their consumption or shift it to time periods in which the grid is less stressed. To empower users with this capability, a first generation of devices, called *Advanced Metering Infrastructure* (AMI), are being installed in households. In the near future, this type of device, apart from improving data collection activities, is envisioned to enable utilities to send pricing signals that alert users when critical periods will occur. In this context, AMIs will be also designed to let utilities implement direct control of demand side management resources. Specifically, the interface through which external entities interact with the AMI, and thus the household, is called *Energy Services Interface* (ESI). An overview of the AMI devices to be installed in households can be found in [Hop08].

Both user participation in DR programs and the involvement of DER devices (including RES) in the generating process is commonly referred to as the composition of a **bi-directional** grid. There is a general consensus that these capabilities can only be achieved through intensive use of latest *Information and Communications Technology* (ICT). Specifically, it is necessary to have an ICT layer that: (*i*) allows users to send detailed information on local processes; (*ii*) enables utilities to inform users about pricing and other conditions; and (*iii*) helps users to develop advanced behaviors, such as coordination and negotiation tasks. In particular, this latter feature aims to create distributed management systems, which may take the form of energy markets.

The goal of the above-mentioned characteristics is to achieve what is commonly referred to as the **Smart Grid**. Accordingly, this is an energy network that, compared to the classical approach, aims to be reactive, distributed, participative and bi-directional. In particular, the European Technology Platform defines it as "*an electricity network that can intelligently integrate the behavior and actions of all users connected to it (generators, consumers and those that do both) in order to efficiently deliver sustainable, economic, and secure electricity supplies*" [ETP06]; while the US Department of Energy describes it as a network that "*uses digital technology to improve reliability, security, and efficiency of the electricity system*" [DOE09].

Given the deep roots that have held the traditional electrical grid in place, the transition to the Smart Grid is conceived as a long-term process that involves legislative, architectonic and technological changes. Furthermore, the excessive dependence of modern societies on energy supply encourages undertaking gradual changes that do not affect the security and reliability of the system.

2.2 Distributed Energy Networks

2.2.1 Concept and architecture

In the medium term, as part of the transition plan to the Smart Grid, it is expected that enclosed areas with a high presence of DER are integrated with the electrical grid, micro-grids being the best-known example of this trend. A micro-grid is commonly defined as an aggregation of loads and distributed sources that operate as one unit capable of producing power and heat [Las02]. From the point of view of the system operator, micro-grids behave like any other point of the network, so they are considered an effective means for the transparent and gradual integration of DER devices into the grid, including renewable energy sources. Figure 2.1 depicts the scheme traditionally used in the literature to illustrate the micro-grids. As shown, this is composed of feeders that can be devoted to specific types of loads. For instance, in Figure 2.1, the two upper ones contain loads that, correspondingly, can be discarded or adjusted to a specific level of consumption; while the bottom feeder is devoted to critical loads that the control system strives to preserve. By separating the devices according to their type, non-critical loads can be disconnected rapidly in case of an emergency or lack of supply. This action is commonly performed by the *Separation Device* (SD), which is officially responsible for facilitating the transition to isolated operation. Specifically, a micro-grid is in islanded mode when it is isolated from the main grid so that it remains operational and functional as an autonomous entity. Furthermore, all communications between the main grid and the micro-grid are performed by a device called *Point of Common Coupling* (PCC). Note that, depending on the context, the scenario depicted in Figure 2.1, instead of being composed of end-use units, may be based on facilities that operate as controllable end-nodes.

As for behavior, micro-grids are said to be *good citizens*, which means they are entities that, by definition, pose no risk to the network and do not add complexity to its management either. In addition, for more advanced phases, it is expected to implement the behavior *model citizen*, by which the micro-grid will also provide ancillary services to the main grid, by either injecting energy when necessary, or limiting its demand when requested.

At a higher level of abstraction, a micro-grid is a specific implementation of an enclosed, autonomous area of the electrical grid. In [PLSW06], the National Sandia Laboratories uses the concept of *cell* to describe a similar structure. In this case, *cells* are defined as set of distributed energy resources that are simple enough to be managed by a single entity based on local principles. Furthermore, the control system is supplemented by a software agent that is responsible for interacting with the neighboring cells. According to the nature of interactions, two organization models are possible:

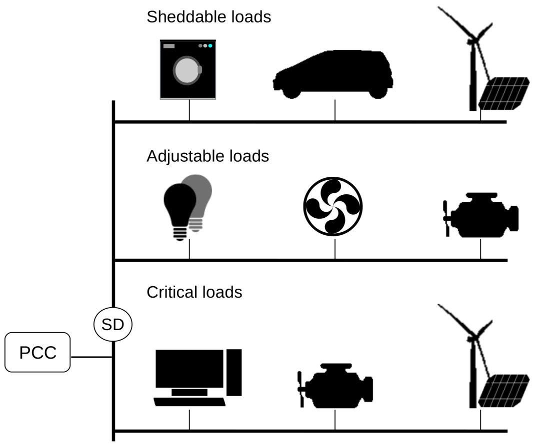

Figure 2.1: Basic scheme of micro-grids.

- *Glob*: A network composed of cells that, following their own interests, negotiate energy exchanges among themselves. The control agent is responsible for negotiating the purchase and sale of energy.
- *Co-op*: A network composed of cells that, besides having all characteristics of Glob cells, are also able to cooperate with each other in order to achieve collective goals.

In turn, a cell of type Co-op, due to its capacity to coordinate with other cells and pursue common goals, can participate as an internal element in other Co-op and Glob cells, thus making it possible to create composite structures. From a practical standpoint, using Co-op cells is the most feasible approach to tackling the development of the Smart Grid, since it allows the definition of goals related to the reliability and quality of the energy supply.

On the other hand, the EU *CRISP* project [ECN06] uses the term *energy cell* [ARP+02] to refer to enclosed, self-managed areas of the distribution network. In this work, one of the most representative characteristics of energy cells is that they can be grouped so that the union of two or more energy cells can make a new cell, thus setting up a structure capable of scaling horizontally and vertically (Figure 2.2). A key difference between energy cells and micro-grids is that units of the former can be other cells, while micro-grids are intended to be composed of generation and consumption entities.

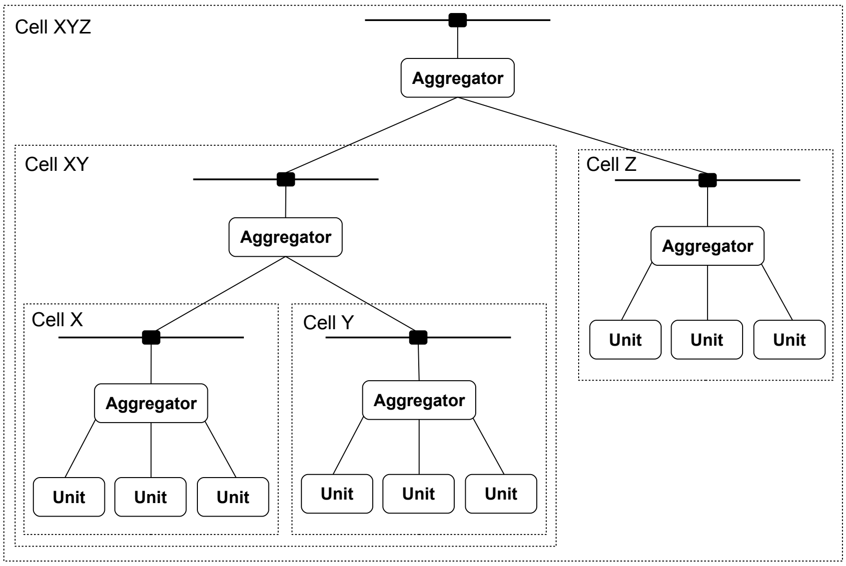

Figure 2.2: Architecture based on the concept *energy cell* of the CRISP project.

As an initial step to implement the energy cell concept into the electrical grid, the CRISP project proposes [CRI02] two hierarchical levels:

- *Level 1*: A cell that is made up of devices belonging to one or more feeders of the distribution network. The boundaries of this type of cells are substations.
- *Level 2*: A cell that arises from grouping Level 1 cells that are connected to the same medium voltage transformer.

2.2.2 **Energy Management System**

As part of its daily operation, distributed energy environments, either they are micro-grids or energy cells, must meet economic, heat load, environmental and legislative constraints. Therefore, apart from the fast electrical control systems, these environments require an intelligent global control system. This is called *Energy Management System* (EMS) and primarily aims to optimize cell's energy cost through planning, coordinating and supervising the activity of all resources [KSLK03]. The EMS works in the secondary control system making short-term plans based on factors such as: conditions imposed by the main grid, specific features of generating units, amount of load that can be modulated and shed, amount of energy that can be stored, energy prices, current legislation, demand estimates and weather forecasts.

It should be noted that the EMS does not necessarily imply the presence of a physical device. The EMS is primarily a concept that may be implemented using the simplest method, such as the hand-control, or the most modern and sophisticated ones, such as distributed systems based on concepts and techniques belonging to the artificial intelligence field. In any case, building an EMS is recognized as a complex task. In practice, the EMS is mostly implemented as a centralized module that is part of a hierarchical control structure with three levels [DH05]:

1. *Distribution Network Operator* (DNO) and *Market Operator* (MO): The DNO is a management system responsible for the operation of the medium or low voltage area that the micro-grid is connected to. Thus, the area of action of the DNO can span multiple micro-grids and utility grids. On the other hand, the MO is responsible for the economic operation of one or more micro-grids.
2. *Micro-Grid Central Controller* (MGCC): After receiving information from the DNO and MO, as well as from internal sensors and components, the MGCC develops action plans and sends commands to the controllable units.
3. *Local Controller* (LC): Each controllable unit of the micro-grid has associated with a LC that is in charge of monitoring its activity and applying the commands sent by the MGCC.

In this scheme (Figure 2.3), the EMS works as an embedded module of the MGCC devoted to the schedule of the local units operation. In particular, the EMS is commonly proposed as a non-linear optimization problem [HTV$^+$04, HAIM07] that includes variables referencing to economic factors and technical characteristics.

However, the aforementioned solution is considered neither efficient nor scalable for medium to large distributed environments because:

i. The computational cost of finding a solution increases exponentially with the size of the model, so that it can easily result in a NP-Hard problem.
ii. Stochastic and nonlinear variables typical of energy units are difficult to model, so they have to be simplified or omitted.

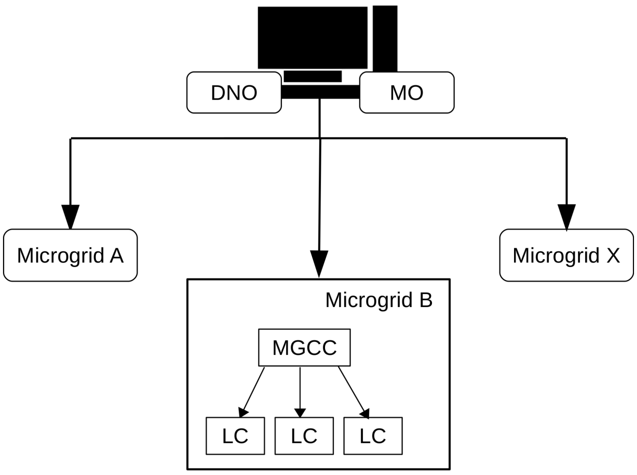

Figure 2.3: Control levels of the micro-grid environment.

iii. The system has low reactivity, since any change in the environment demands restarting the optimization process.

iv. The system provides a low level of autonomy to users, who are limited to expressing their intentions through prices or utility functions.

Even though there are also works based on neural networks [CPPS06] and fuzzy logic [KDAP12, LSM09], adopting a centralized approach for implementing the EMS is considered unsuitable because it grants neither autonomy of users nor the reactivity and flexibility required by the Smart Grid. On the contrary, distributed control solutions fit better with an environment like the Smart Grid, which aims to be bi-directional, distributed, intelligent and reactive. In response to this demand, the EU CRISP project puts forward the **Supply and Demand Matching** (SDM) management model [KCKA04], whereby entities owning generation and consumption resources can dynamically bargain exchanges of energy blocks so that the network is continuously balanced. The SDM model stands out for providing autonomy to producers, unlike techniques such as DSM (*Demand Side Management*) and DRR (*Demand Response Resources*), in which only authority nodes and consumers have capacity to act.

In essence, the SDM model proposes the creation of micro-energy markets in distributed energy contexts such as cells and micro-grids. On a smaller scale, they emulate the mechanics of wholesale energy markets: through negotiations each node decides the amount of energy it produces and consumes, and for how long the action is carried out. Micro-energy markets are conceived as being highly reactive, instantiated on demand, and with a short time horizon (usually shorter than 15 minutes).

The SDM model, as well as many other solutions related to the Smart Grid, require placing an autonomous piece of software at each node, which has the mission of: (*i*) representing the interests of users in micro-energy markets; and (*ii*) coordinating with other nodes in order to meet collective goals. Intelligent agents are accepted as the most suitable technology to address this challenge. However, the lack of stable standards for a noticeable period of time and the need to make assumptions about future scenarios, have resulted that many studies that propose software agents for the control of the Smart Grid do not share a common vocabulary. In order to proceed with our study, this work assumes the presence of the basic components described below, which are usually found in the literature (Figure 2.4):

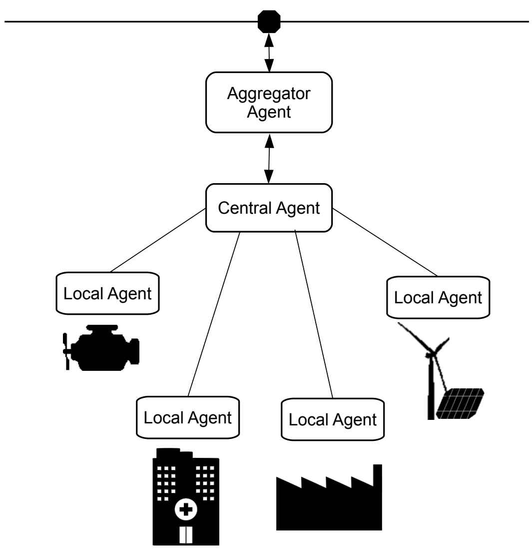

Figure 2.4: Common roles that play software agents in energy cells.

- *Local agent*: This represents production, consumption and storage entities throughout the management process. The main tasks of local agents are to negotiate on behalf of customers, sending commands to the local devices, monitoring their activity, and sending updated information to the authority nodes.

Local agents are usually planned to be run in the control device of the customer's facility.

- *Central agent*: It is a system agent that, depending of the level of decentralization of the solution, is in charge of supervising and/or controlling the operation of the micro-grid or energy cell. This role is essentially the agent-based version of that of central controller in the traditional approach.
- *Aggregator agent*: It manages the interactions of the cell with the main grid and other external entities, which in turn can be other cells or micro-grids.

The participation of the central agent in the clearing process varies according to the decentralization level of the solution. Three main tasks are distinguished along the state of the art:

- *Management*: The central agent is responsible for developing and controlling the actions plans of all the entities. These plans are carried out according to the information sent by the local agents regarding both the state of the units under control and the users' preferences.
- *Supervision*: The central agent monitors the activity of the local agents, which in this case are the entities responsible for drawing up the actions plans. The central agent may refuse or intercede on both the plans and objectives in order to ensure that conditions related to the efficiency, security and reliability are met.
- *Services*: The activity of the central agent is limited to providing support through functions and data services. In this regard, FIPA protocols [FIP96] define set of services intended to facilitate typical tasks in multi-agent systems, such as locating and registering agents. Also, in order to make reliable plans, local agents will likely require data services such as demand estimates and weather forecasts. As for the business logic, they are also necessary functions that control the market's life-cycle.

Furthermore, regardless of the type of approach, the central agent is commonly proposed to record the activity of the system's components, and confirm that local agents behave in accordance with the agreed plans and goals.

Environments in which the central agent performs the clearing process are essentially centralized solutions [HDT$^+$05, OJ05, KWK05, FTNY08], so that the functions of the central agent are practically identical to those of the Micro-grid Central Controller (MGCC, [HTV$^+$04]), which is a physical device designed to take on the entire control of the micro-grid. When the central agent works as supervisor [DH04, DH05, Arn00], the solution gains in decentralization, as it arises from the interaction of local agents. However, in this case the outcome still requires the approval of the central agent, which may be programmed to look out for parameters such as grid stability, power quality, supply security and efficiency. Finally, when the tasks assigned to the central agent (when necessary) are limited to providing ancillary services, the solution can be considered fully decentralized [AB00, RPT07, PFR09, LKG05, Jia06, BCG$^+$98, PLSW06].

Much of the literature tends to include the aggregator agent as a subsystem of the central agent. This work represents these two figures separately though because they actually work in well-differentiated functional areas that, due to their complexity, require individual analysis. As mentioned before, the aggregator is in charge of managing the interactions of the cell with the context. The aggregator may receive instructions from the system operator, sends information to it about the local devices, and manages the exchange of energy with the main grid and surrounding cells. The presence of the aggregator agent is common in environments in which local agents are able to coordinate and cooperate with each other. Internally, the aggregator communicates all the information and instructions to the central agent, which is responsible for processing them.

To sum up, the EMS, when implemented in a distributed manner, is a multi-agent system in which software agents, representing local nodes, interact and coordinate between them in order to balance the system and accomplish both particular and collective goals. This type of implementation shows that, indeed, the EMS is more conceptual than physical, since the management here arises as result of the communication and coordination of independent software agents.

2.3 Standards for the energy management

The most important effort to standardize the Smart Grid comes from the *National Institute of Standards and Technology* (NIST). This aims to guide the development of a framework that includes standards of systems, devices and procedures

[NIS12]. In order to support NIST in this task, the *Smart Grid Interoperability Panel* (SGIP) was established in late 2009, which, in collaboration with external organizations, aims to define requirements for essential communication protocols and other common specifications.

As a first step in the race to the Smart Grid, SGIP has identified key areas for which standards should be developed. Furthermore, a small group of these standards was considered highly important, being classified as *Priority Action Plans* (PAP). The PAP09 is specifically devoted to the development of DR programs, thus recognizing the importance of this area in the short-term future of the Smart Grid. Much of the work devoted to this plan has focused on the definition of the OpenADR standard, which is supported by the information and communication model described in the Energy Interoperation standard developed by the collaborating organization *Advanced Open Standards for the Information Society* (OASIS). The basic concepts of both standards are briefly described below.

2.3.1 The Energy Interoperation standard

The goal of the *Energy Interoperation* (EI) standard from OASIS is to define messages to communicating prices, reliability and emergency conditions. Formally, the standard is said to describe "*an information and communication model to coordinate energy supply, transmission, distribution, and use, including power and ancillary services, between any two parties, such as energy suppliers and customers, markets and service providers*" [OASa]. It is important to highlight that, in the architecture defined in the EI standard: (*i*) interactions are always possible between any pair of actors; and (*ii*) an actor can participate in many interactions at the same time. The standard adopts a services-oriented approach and is agnostic in relation to the technology used to carry the messages. As for the local devices, facilities must be provided with communication interfaces such as that described in [Hol09]. Specifically, the point of communication whereby nodes offer and consume services is the ESI.

The information and communication model defined in the EI standard is intended to facilitate collaboration in energy use. *Collaborative Energy* stands for the management of energy using cooperative mechanisms. In addition, when there are market interactions, the management model is referred to as *Transactive Energy*. In this scheme, parties buy and sell energy using tenders that, if accepted, result in transactions (Figure 2.5). In a transaction, a party can take on the role of buyer or seller. Normally, a generator will be on the seller's side of the transaction, and an end-use customer on the buyer's one; although nothing prevents them from swapping these roles. As for the negotiation process, the tender that initiates the transaction can be sent by any of the parties.

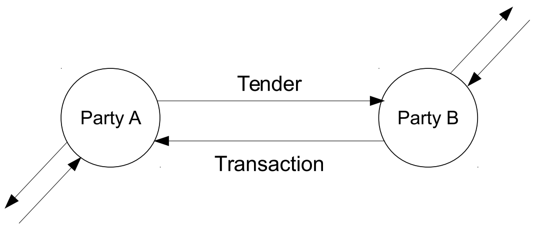

Figure 2.5: Parties interacting using tenders and transactions in the EI standard.

Apart from the Transactive Energy model, the EI standard also defines a structural model for interactions typical of DR programs, which consists of event-based dispatch of resources. The model is principally based on the definition of two roles: *Virtual Top Node* (VTN) and *Virtual End Node* (VEN). A VTN can interact simultaneously with many VENs, while VENs are not allowed to interact directly among themselves. As in any interaction of the EI standard, parties may participate in many interactions concurrently. In this case, a node may implement both interfaces, playing the role of VTN in some interactions, and the role of VEN in others.

In the common use case, VTNs are intended to be authoritative nodes, such as the DSO or the Micro-grid Operator, while VENs are intended to represent generation and curtailment resources. Thus, VTN nodes usually send DR signals and requests for information to VENs. The nodes that implement both interfaces are usually aggregators.

Figure 2.6 illustrates how the combination of pairwise interactions of VTNs and VENs enables the implementation of complex structures. The graph could model a DR event initiated by the system operator, which in this case is represented by the node A. Initially, the event is sent to the first-level nodes B and C, which work as aggregators. In a real case, they could represent the controller of a micro-grid, a factory, a smart building or a floor of a building. In turn, the second-level node E wrappers the nodes F, G and H, while the node C wrappers the node H. These are all end-nodes. They could represent micro-grid devices, HVAC units, machines or floors of smart buildings. Note that aggregators are not required to re-send the same signal they receive. Actually, they can process it and generate a new set of signals which, from the point of view of the higher level, are usually expected to yield the same result.

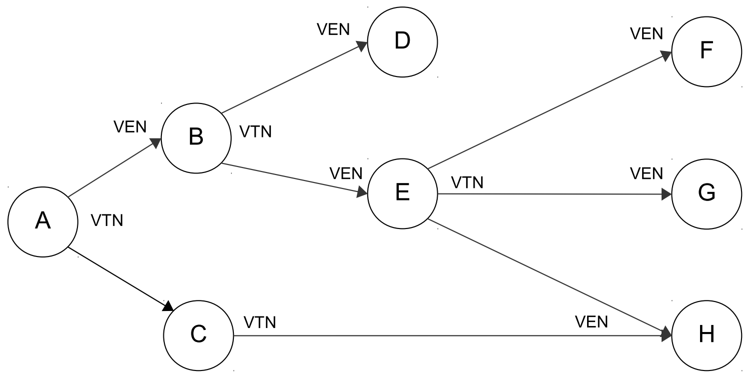

Figure 2.6: Example of interactions between VTN and VEN nodes.

The EI standard is not intended to define all the entities and messages that would require a real scenario. Actually, the concept of real scenario is still vague, since there is a large spectrum of possible contexts and collaborative mechanisms. Thus, the EI standard has been designed to be complemented by other standards. In this regard, it principally relies on the general purpose standards *Energy Market Information Exchange* (EMIX) [OASb] and WS-Calendar [OASc]. EMIX is focused on the definition of entities that represent products, quantities, and prices; while WS-Calendar is a specification to communicate schedules and intervals.

2.3.2 The OpenADR standard

Following the California electricity crisis of 2000 / 2001, the California Energy Commission decided to fund a research program to develop management systems that provide reactiveness to the power grid and are capable of responding dynamically to energy prices. The most immediate consequence was the creation of the *Open Automated Demand Response Communications Specification*, also called OpenADR [OAD], in 2002. OpenADR is defined as a "*communication data model which facilitates information exchange between two end-points, the electricity service provider and the customer*" [OAD12].

In contrast to the usual approach of DR programs, OpenADR was designed to interact with local facility controls that carry out fully automated actions in response to DR signals. As a result, no manual intervention is necessary to handle price signals or curtailment events. Automating response of facilities makes it possible to implement programs that have lead times of seconds or minutes. This scheme is commonly known as fast DR. In contrast, traditional DR programs, also known as slow DR, are based on events that are scheduled significant time before they are run, such as a day ahead. One of the most important benefits of fast DR is reactiveness: it makes it possible to continuously monitor parameters such as energy prices, and to translate them into control actions that follow a predefined strategy. This feature is a valuable resource for the system operator in order to maintain a reliable electric service and avoid high electricity prices.

The version 2.0 of OpenADR has been designed on the basis of the model defined in the EI standard. In specific, in the OpenADR networks the nodes are divided in two groups: (*i*) nodes which publish and transmit information about events to other nodes (e.g. utilities); and (*ii*) nodes that receive and respond to that information (e.g. end-users). Following the terminology of the EI standard, the nodes belonging to the former type are VTN nodes, while the latter are VEN nodes. The role of a VTN node in OpenADR 2.0 is to communicate grid conditions to entities that control demand side resources; while the role of a VEN node is implemented by a producer or consumer of energy that listens and reacts to DR signals sent by the VTN node to which it is connected. In accordance with the EI standard, communications always occur between a VTN node and one or more VENs. There is no peer-to-peer communication, meaning that VTNs do not communicate with other VTNs, and likewise VENs do not communicate with other VENs.

In the standard, the technologies proposed to transmit the messages are HTTP and *Extensible Messaging and Presence Protocol* (XMPP) [XMP]. In the former case, the mechanism used to transport the messages is *Simple Object Access Protocol* (SOAP) [W3C]. The main drawback of this technology is that it requires the side that receives messages to run a web server. On the other hand, XMPP is a P2P protocol whereby nodes are able to initiate bidirectional communications without installing additional software.

With the aim of accommodating all kinds of devices and thus expanding the adoption of the standard, OpenADR 2.0 defines three levels of support: (*i*) *2.0a*, minimal support; (*ii*) *2.0c*, full support; and (*iii*) *2.0b*, intermediate level support. The simplest level (2.0a) is intended to accommodate devices with limited computational resources, such as thermostat and other end-use units. On the other hand, the profiles 2.0b and 2.0c are targeted to more complex devices, such as aggregators and scheduling nodes, which are supposed to include capabilities typical of information and communication systems. The relation of the standards and profiles described so far is illustrated in Figure 2.7. As shown, OpenADR 2.0 is a subset of the EI standard, and defines three profiles, which are compatible services subsets.

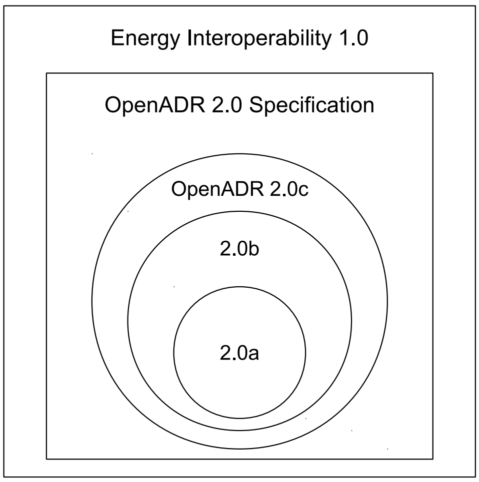

Figure 2.7: Profiles of the OpenADR 2.0 specification.

An important characteristic of the profile 2.0a is that it only supports signals of type *simple*, which are basically defined by using one of the following values: *normal*, *moderate*, *high* and *critical*. The value *normal* means that no restrictions are imposed, so the node can consume as usual. The values *moderate*, *high* and *critical* represent three restriction levels of consumption, from lowest to highest, to be defined by the system operator. In a normal context, all nodes would share the same levels' definition, which means that all of them should react by applying the same set of actions when they receive a specific value. The definition of the levels may consist of switching off devices or limiting their demand.

On the other hand, the profile 2.0b supports a wide range of signal types. Among them is the *delta* type, through which VTNs can define the exact amount of load that VENs must discard. This profile is intended to be used by aggregators and more advanced nodes.

2.4 Challenges of using smart local devices

In the Smart Grid, using distributed smart local devices as control system implies adopting a model similar to the one depicted in Figure 2.8. Here, agents operate within local devices and are responsible for: participating in local energy markets in which they develop planning and coordination tasks, interacting with the user, monitoring generating and consumption devices, and accessing external data services necessary to draw up actions plans. Even though intelligent agents are theoretically capable of performing all these functions, this model entails architectural and technological challenges that have not yet been analyzed.

2.4.1 Architectural challenges

Contrary to what is often stated, in practice running a software agent within a local device similar to an AMI does not configure a flexible and reactive control system such as that required by the Smart Grid. In particular, under such a scheme, due to the complexity and reactivity of the grid, as well as the reliability it requires, software agents would need to be updated or debugged periodically, making the maintenance of medium and large networks into a slow and expensive task. In addition, this problem worsens when the device is also proposed as a means for users to set their preferences and monitor their units. The CRISP project has dealt with the latter approach by installing a mini web server into the local device. However, this approach has the following disadvantages:

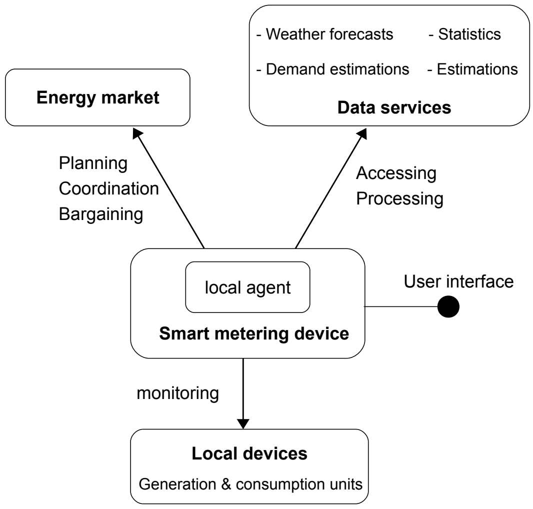

Figure 2.8: Tasks and interactions assigned to the smart local devices in distributed control schemes for the Smart Grid.

- Embedded web interfaces are poor and limited. They are not the best option for monitoring the units or setting users' preferences.
- Users must configure their personal computers in order to obtain remote access to the local control device. This new connection must coexist with the local connection to Internet.
- The control device becomes more expensive and complex due to the need for installing a web server and a new network interface.

It is therefore necessary to design new solutions that insure software agents can be installed and updated dynamically, and also provide a better integration with the common user devices, such as computers, smart phones and tablets.

On the other hand, both micro-grids and energy cells are infrastructures that are compliant with the EI standard as long as they do not allow direct communication between *sibling nodes*. In DENs, two nodes are considered siblings when they receive orders from the same node. To allow direct communication between them would violate the condition that states that a node of type VTN can only interact with one or more nodes of type VEN, and that a node of type VEN can only interact with VTN nodes. Therefore, in order to meet the OASIS standard specifications, software agents of both micro-grids and energy cells are only allowed to communicate with nodes that play the role of aggregator or central controller (Figure 2.9). This is a serious restriction which somewhat contradicts the spirit of multi-agent systems. Indeed, in the absence of new solutions, this constraint practically forces the adoption of solutions with a significant degree of centralization, and prevents many of the functionalities and advantages that the installation of software agents may offer to distributed energy environments.

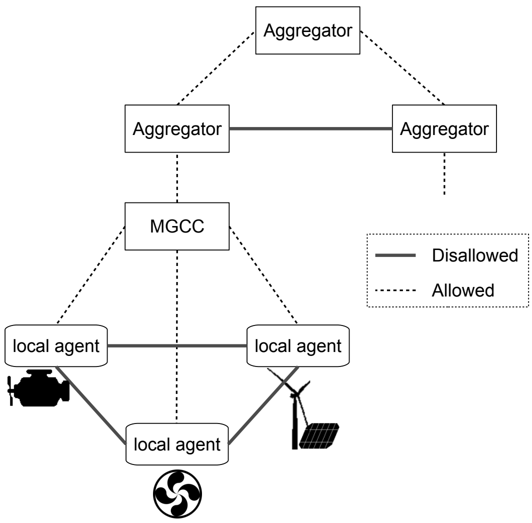

Figure 2.9: Disallowed interactions between local nodes in energy cells and micro-grids.

2.4.2 Technological challenges

In order to plan the activity of the production and consumption units, local agents must have access to external data sources such as weather forecasts, demand estimates and energy prices. This information is essential for bargaining in local energy markets, and is expected to be offered by specialized companies in the sector through web services. However, accessing external services, processing their information and carrying out the planning and coordination processes typical of the energy markets are tasks that may be too demanding for local devices with limited resources. Adding this type of functionality would add complexity to the AMIs, thus raising their price and making them more difficult to maintain. This condition is clearly undesirable, since these devices are intended to be installed massively. In particular, three important data services have constantly been proposed for the correct functioning of the Smart Grid:

- *Demand estimates*: These are necessary in order that the system operator knows when curtailment events are necessary, and likewise nodes know how much demand they must shed or shift. To supply this type of service implies having databases that contain information about each user's consumption throughout the year.
- *Weather forecasts*: Consumption depends highly on factors such as atmospheric temperature. In winter, water-heaters and HVAC systems represent an important source of consumption, while in summer air conditioning systems are most important. Therefore, to obtain accurate demand estimates, it is necessary to have accurate weather forecasts.
- *Energy prices*: The demand for energy, as well as the reaction to DR events, may depend on the energy prices. If it is possible, users may be willing to configure their consumption level according to price levels.

Multi-agent systems are a difficult matter. Although intelligent agents are continuously being proposed for the implementation of distributed management systems, the truth is that, in practice, developers and researches tend to choose more practical solutions for real cases. Actually, despite the high number of research studies that propose software agents for the management of distributed virtual environments such as grid computing and P2P networks, at the present time solutions based on intelligent agents are not widely adopted. In fact, the practical application of agency theory is mainly focused on the domain of processes and server applications, while solutions which connect agents with users are limited (see 3.1.2). This lack of success is partly due to the complexity that arises from solutions based on agents, and the absence of models adapted to the real habits of users, who are increasingly demanding transparent and simple solutions that avoid technological details.

2.5 Lessons learned from similar fields

2.5.1 Peer-to-peer networks

In particular, *Peer-to-peer* (P2P) networks [ATS04] aim to facilitate the exchange of resources between peers that, in theory, can be considered equals in terms of functionality. Their implementation has traditionally been focused on exchanging files, *Napster*, *Gnutella* and *eDonkey* being the best-known cases. All these networks include ancillary services that facilitate processes such as the interconnection of peers, the search for resources and the classification of contents.

One factor that has proven decisive in the success of P2P networks is the topology (Figure 2.10). In this regard, there are three main options [Sch01]:

- *Centralized*: This is the simplest scheme. Peers connect to centralized servers in order to access special functionalities. Among them are the search for resources, and user registration. Note that resources (files in most of the cases) are still exchanged directly between peers. The most representative example of this type of network is Napster.
- *Decentralized*: All services, including the registration of new peers and the search for contents, can be carried out in each node. The implementation of these tasks is performed by sending request messages to the closest neighbors. From there, messages are recursively propagated until reaching a maximum iteration depth. The most representative network using this topology is Gnutella.
- *Hybrid*: This topology uses special nodes, called ***supernodes***, that, for a limited section of the network, work out as entry points for users, indexing all their contents, and processing search requests. Supernodes are connected between themselves so that they can exchange information about the network and its contents. Note that, by sharing this information, supernodes are able to search for contents in the whole network. eDonkey is the most best-known implementation of this type of networks.

The main drawback of the centralized topology is having a single point of failure, thus being too vulnerable to attacks and prosecution. As a case in point, it took only one day for the authorities to shut down Napster. Alternatively, there are

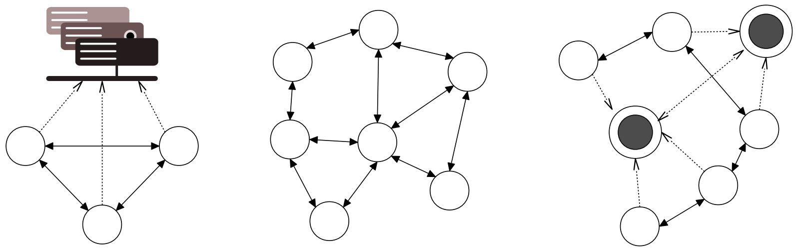

Figure 2.10: Main types of P2P networks.

decentralized networks. However, the lack of a full index of available resources in the latter type has proven to make the searching process ineffective. In addition, these types of networks have proven to be hard to scale and maintain. In order to overcome these drawbacks, in networks such as Gnutella, nodes have emerged that are able to handle large numbers of connections and take on special functionalities on behalf of other nodes, such as finding resources. In practice, this approach makes Gnutella resemble hybrid networks, since special nodes behave much like supernodes. As a matter of fact, as shown in Figure 2.11, the actual topology of Gnutella is similar to a hybrid one, thus reinforcing the thesis of the latter approach. Actually, the hybrid topology is the one most widely used in practice, having many successful implementations, and also having proven to be the most efficient for exchanging resources. Its success relies largely on the assumption that all nodes are not equal: they are not actually peers, since in practice they have different characteristics, including computing power, bandwidth and quality of service. Therefore, it is natural that, in order to improve the overall system performance, there are some nodes that have to take more responsibilities than others.

As for the Smart Grid, since the OASIS and NIST standards leave the door open to the installation of nodes with different profiles, it is advisable to study the benefits that may arise from the installation of nodes which are more powerful than those envisioned so far.

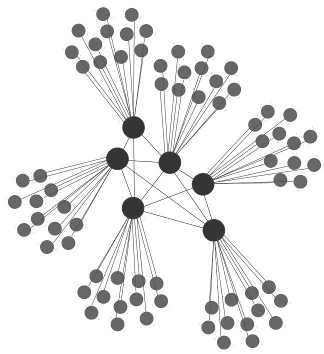

Figure 2.11: Structural pattern of the Gnutella network.

2.5.2 Grid computing

The aim of a computational grid is to create the image of a powerful computer through the interconnection of heterogeneous interconnected systems [FKT01]. This goal is very similar to that pursued by the Smart Grid, which strives to build a large generation system from the joint production of distributed, small energy resources.

The first computational grids were ad-hoc solutions implemented from scratch. As a result, these were difficult to replicate in other target environments. The second generation was characterized by the creation of frameworks and tools that facilitated the implementation of computational grids, as well as an application ecosystem around them. Among the best-known frameworks were *Legion* [GWTLT97], *Condor* [TTL05] and *Gridbus Toolkit* [FK97]. However, early versions of these frameworks were monolithic, hard to scale and with little capacity to connect to external middleware layers. As a result, many island grids emerged in the US with architectures that must be defined as too specific as they were principally intended to finding practical solutions, neglecting important features such as scalability and interoperability. As a matter of fact, in reference to the roughness of the solutions, this stage is commonly described as "*big irons and fat pipes*" [GDR04].

The third generation of computational grids was born embracing the services orientation [Fos05]. This replaced the concept of *resource* with that of *service*, with the result that nodes of the grid actually offer and consume services that mask resources. Also, in order to increase interoperability, this new generation of computational grids promotes the adoption of open standards. In this regard, the Globus Toolkit team and IBM contributed to the creation of *Open Grid Services Architecture* (OGSA), which is considered the *de facto* standard for the implementation of computational grids [FKT04].

In addition, as occurs in the Smart Grid, there is a school of thought that touts the virtues of using market-based mechanisms as management system. This approach is known as *grid economy* [BAV05], and proposes to switch to a model in which clients are autonomous entities who try to defend their own interests and goals through negotiation systems. This approach can be defined as *user-centric*, while the traditional model that looks to improving the global efficiency of the system is defined as *system-centric*.

Compared to other technological fields that face the challenge of sharing and coordinating distributed resources, the grid computing community has been praised for its ability to achieve valid solutions. Although these solutions are often described as being *rigid*, it is also true that a direct approach has proven to be effective for achieving operative systems.

The grid computing community has also studied the benefit of integrating software agents throughout the architecture [FJK04]. These are mainly proposed to provide flexibility and automate the management tasks, and act on behalf of users in market-based environments. However, the truth is that the presence of software agents in real systems is scarce, possibly because this type of solution adds a new level of complexity in software development, requiring knowledge of the field of artificial intelligence.

In conclusion, the most important lessons learned from the grid computing experience are to:

i. Embrace a services-oriented approach;

ii. Intensify efforts dedicated to the definition and adoption of standards;

iii. Reach a compromise over the need to find practical solutions; and

iv. Devote more research efforts to achieving solutions based on software agents.

2.5.3 Virtual organizations

In the business world, enterprises are also experiencing challenges that, in essence, are similar to those addressed by the computational grids and the Smart Grid. In particular, due to the growing trend towards specialization, business opportunities are increasingly fulfilled by temporary coalitions of enterprises that cooperate and share knowledge, resources and competences. This type of coalition is known as *Virtual Enterprise* (VE); a concept that usually arises when individual enterprises do not have the resources to achieve a specific goal acting on their own, or to do so profitably [MFPF01]. In order to take advantage of this vision and learn from the experience gained in this field so far, the second generation of computational grids started using the concept of *Virtual Organization* (VO), which aims to apply the concept of VE in environments that are essentially technological. Specifically, VOs are defined as temporary coalitions of distributed entities that collaborate and share resources to meet global and individual goals making intensive use of new information and communication technologies [NT07]. This approach arose in response to environments that are increasingly changing, agile and distributed, in which partners look for alliances that help them to achieve new goals, increase their competitiveness and reduce risks. Societies that are classified as VOs commonly share the following properties:

- They are specifically created for meeting a temporal business opportunity.
- They have a strong dependence on ICT.
- They do not require the partners to be collocated in order to carry out the assigned tasks.
- They are capable of adapting their structures to the needs of the context.
- They make an intensive use of cooperation mechanisms in order to achieve the defined goals.
- They are composed of autonomous entities that, besides pursuing global goals, strive to meet their own goals.

Distributed energy networks, particularly when the management system is based on a distributed mechanism, meet these properties: their activity is envisioned as being supported by multiple autonomous units that cooperate in order to guarantee global goals (such as reliability and security of supply) and particular goals (such as exchanging energy in a profitable way) by using the latest information and communication technologies. Therefore, distributed energy environments can be conceptually considered VOs, thus being in good position to learn from important undertakings in this field in recent years. However, in difference to the attention paid to grid computing, the literature seems to have ignored this important example.

Specifically, the experience of VOs warns us that the automation of the entire life cycle of a VO is a complex task, which in practice requires specific solutions, and usually the intervention of human operators. This experience therefore shows that, despite the remarkable progress in computer technology, the creation of distributed virtual environments inhabited by autonomous entities is a difficult task which at present requires the implementation of ad-hoc solutions, even the supervision of human actors. In particular, the main challenge faced by VOs is the implementation of the *creation stage* [CM06], which must accomplish tasks such as: opportunity identification, action plan designing, suitable partners selection and tasks assignation. In addition, the Smart Grid poses typical challenges of open and reactive environments, such as communication and coordination between heterogeneous agents, and the implementation of trust mechanisms that help to avoid the risk that the presence of agents with unknown reputation creates.

To overcome the complexity of the *creation* stage, VO researches have proposed creating a specialized environment called *Virtual Breeding Environment* (VBE), which is a stable limited cluster composed of well-known and capable partners that maintain long-term relationships [CMA03]. A VBE imposes on the partners the use of common technological infrastructures, ontologies, communication protocols and social conventions. Furthermore, a VBE authority certifies the skills of each partner, thus proving that it is suitable for being part of VOs in the future. All these conditions are intended to configure a safe, reliable and normalized environment that facilitates the dynamic installation of VOs. As for its drawbacks, it must be noted that a VBE is a semi-closed environment that, to some extent, lacks flexibility and restricts participation.

NIST and OASIS standards cover some of the features required to VBEs. They define the architecture, the communication protocols and the technologies to be used, including all issues related to the security of the network. However, in order to achieve a fully operative system based on autonomous, self-interested agents, it is still necessary to establish the negotiation algorithms and social conventions through which agents must behave under normal and exceptional situations. In this regard, VBE experience shows that the more defined and limited the context is, the easier it is to implement an effective solution. Here, the challenge of the Smart Grid community is to establish a well-defined framework that also preserves the autonomy of software agents and promotes the participation.

En mi largo trato con el mar aprendí que lo más natural del mundo son los cambios.

"*La obra*", Adolfo Bioy Casares.

Quién sabe. A mí me parece que los peces ya no quieren salir de la pecera, casi nunca tocan el vidrio con la nariz [...] Chestov había hablado de peceras con un tabique móvil que en un momento dado podía sacarse sin que el pez habituado al compartimento se decidiera jamás a pasar al otro lado. Llegar hasta un punto del agua, girar, volverse, sin saber que ya no hay obstáculo, que bastaría seguir avanzando.

"*Rayuela*", Julio Cortázar.

CHAPTER 3

Agency Services

As described in Chapter 2, the energy management of DENs is often envisioned in the form of a multi-agent system. Under this approach, end-nodes are represented by software agents in local energy markets in which they plan and conduct the action of the production and consumption units. The origin of this approach is based on the theoretical properties of intelligent agents, which are formally described as entities capable of providing autonomy, intelligence and reactivity in distributed environments. However, the proposals based on this idea have given little attention to the fact that intelligent agents, contrary to the enormous expectations built up about them for more than ten years now, have actually had little practical impact on technological areas that also seemed suitable for them. Some remarkable examples of these areas are computational grids, P2P networks and the multiple types of virtual societies created around Internet.

In this light, this chapter first discusses the limited success of intelligent agents in the practical field. To this end, the question "*Where are all the intelligent agents?*", which was recently put to the community by an authority on the subject, is used as starting point. The truth is that the complexity of typical artificial intelligence solutions, together with the lack of knowledge on the subject, poses insurmountable barriers for teams facing multidisciplinary challenges. As a matter of fact, authors usually propose solutions in which customers must perform development tasks, or are involved in tedious installation and configuration processes. As will be explained, this approach is contrary to the type of product that, at the moment, prevails in the world of software. Specifically, the cloud computing paradigm seems to have found the right key by delegating many complex tasks, including managing resources, to third-party companies, so that users only need to worry about using the applications.

This chapter, inspired by the principles of the cloud computing paradigm, puts forward the *Agency Services* model, which is as a novel architectonic solution for the practical implementation of multi-agent systems in distributed environments. This suggests a paradigm shift in designing distributed agent-based solutions. Secondly, the text examines in detail the adaptation of the model to the particular case of distributed energy networks. Specifically, it discusses the characteristics of the novel approach, the benefits that it would bring, and, not least of all, it studies the compatibility of the new model with the current OASIS and NIST standards.

3.1 Intelligent agents

3.1.1 Concept

An *intelligent agent* is an autonomous entity capable of developing flexible action planning in a certain environment in order to achieve well-defined goals [WJ95]. The main features of intelligent agents are:

- *Goal orientation*: Agents are designed to achieve specific well-defined goals that include: *global* or *collective* goals related to the environment where they are embedded; and *particular* goals related to the interests they represent.
- *Autonomy*: Agents are capable of deciding which actions must be applied to achieve specific goals.
- *Social ability*: Agents communicate with each other and with human actors through interfaces and standard communication languages.
- *Reactivity*: Agents adapt their response to changes in the environment they inhabit.
- *Proactivity*: Agents develop action plans and take initiatives related to the global and particular goals to be realized.

By extension, a *multi-agent system* [Woo09, ACC$^+$99] is an environment where multiple agents are coordinated or compete in order to achieve particular and collective goals. On the other hand, a *software agent* is a software entity which functions continuously and autonomously in a particular software environment, often inhabited by other agents and processes [Bra97]. As intelligent agents, software agents are expected to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment. Furthermore, it is noted that this definition highlights their ability to act autonomously, without requiring constant human guidance or intervention.

3.1.2 Where are all the intelligent agents?

Contrary to what was expected, intelligent agents have not become a widely adopted technology, meaning there is a significant gap between theory and practice. As a matter of fact, important voices in the field, in an effort to generate debate on this issue, are openly asking *"Where are all the intelligent agents?"* [Hen07, Dra09]. Although certainly many such agents may be acting everywhere [ML07], masked in internal tasks of larger processes and systems, it is also true that the development of solutions explicitly oriented to software agents has a minority status. The absence of intelligent agents is particularly noticeable on the users' side, where they have the potential to guide and represent users in the virtual sites that have emerged in the Internet, such as those dedicated to holding auctions or buying and selling products. Also, software agents have failed to be implemented in thriving technological fields that, due to their characteristics and behaviors, actually need them, such as grid computing, peer-to-peer networks, semantic web services and virtual organizations. This absence is due mostly to the complexity that designing and implementing solutions based on software agents involves. Such complexity is important in multi-agent systems, where autonomous intelligent agents are supposed to interact and communicate with each other in order to achieve both particular goals and collective goals.

In order to address the gap between the actual use of this technology and the expectations created about it, significant work has been dedicated to building programming frameworks and software tools. However, this approach requires users to implement their own agents, and this is not in line with the type of solution that predominates in the software world. On the contrary, users are increasingly demanding solutions that free them from all tasks that do not have to do with the simple use of the applications. This condition is even more important in the case of multi-agent systems, since the problems involved are complex in nature, including tasks related to planning, cooperation and negotiation in distributed environments which are shared and cohabited. Aware of this complexity, some research projects opt for installing an agent on the server of the target site [San02]. However, this approach is limited because: (*i*) agents cannot behave neutrally; (*ii*) local resources are not monitored; and (*iii*) the interaction with the user is always carried out as part of a web session.

The lack of practical success of intelligent agents in virtual societies suggests that new efforts have to be made in the design area. Otherwise, as some authors point out [FJK04], intelligent agents are at risk of becoming a largely theoretical subject which fails to translate much of its progress into practice. Other areas of computing, such as web services and grid computing, have successfully taken the opposite course. They have focused their work on the development of practical infrastructures and promote communication standards that meet immediate needs. In order for intelligent agents to provide real solutions for the new virtual environments that are emerging as result of ICT, it is necessary to design new models in line with the habits and expectations of today's users: simple solutions which are able to represent the interests of clients without limiting their autonomy.

3.2 Cloud computing

The method of marketing the software and the resources around it is rapidly changing. New advances in ICT, and the price depression of the storage and processing resources, have led to the traditional model, based on direct software sales, being gradually replaced by a model based on the rental of resources to remote data centers that are accessed through service-oriented protocols. This new way to market and consume computing resources is known as *cloud computing*; a paradigm that, despite being in its infancy, has already been widely adopted with considerable investments, by major companies such as Google, Microsoft, IBM or Amazon. Formally, the term *Cloud Computing* is defined as "*a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be provisioned rapidly and released with minimal management effort or service provider interaction*" [MG11].

Cloud computing is generally considered to be the materialization of the *Utility Computing* concept conceived decades ago, which proposes the leasing of computing resources to remote clients. In the cloud computing model this idea is built on three roles [ZCB10]: (*i*) the *infrastructure provider*, which is responsible for leasing resources on demand using virtualization technologies, along with security and balance policies; (*ii*) the *service provider*, which rents resources to one or more infrastructure providers to provide new functionalities as services; and (*iii*) the service clients, who are the final consumers. In practice, the infrastructure provider and the service provider are usually represented by the same entity. In particular, the cloud computing model is characterized by resources being allocated and released dynamically according to customer needs, who pay for their use, but not necessarily through flat fees or long-term contracts. Through means of a *Service Level Agreement* (SLA), customers and suppliers reach an agreement concerning aspects of required resources, pricing and quality of service (QoS).

Depending on the type of resource being offered, the services are classified as:

- *Infrastructure as a Service (IaaS)*: This offers processing and storage resources.
- *Platform as a Service (PaaS)*: This offers a software development framework that uses resources that are also commonly provided by the same entity in the role of IaaS.
- *Software as a Service (SaaS)*: This offers software applications that users can run without having need to install them. This type of software application is commonly based on the services of an external IaaS.

Regarding SaaS, one of the most outstanding virtues of the new paradigm is that it frees users from all operations that have nothing to do with merely using the application. In general, the benefits of using the software as a service are that: the installation and maintenance processes are simplified, applications are accessible via standard Internet protocols, configuration and application data can be stored in the cloud, users pay for specific features, and applications are enriched with functionalities related to sharing information online with other users and applications.

In general, the advantages attributed to the cloud computing model are that:

1. It eliminates the need for large upfront investments in infrastructure of uncertain profitability.
2. It saves costs by relying on a pay-per-use pricing model.
3. It scales dynamically by allocating resources according to demand for services.
4. It transfers technical risks and maintenance costs to the owner of the infrastructure.

3.3 Agency Services

3.3.1 Description

Following the success of the cloud computing paradigm, the idea behind the *Agency Services* (AS) model is to transfer much of the complexity of developing and managing software agents to third-party, specialized companies. In short, the AS model proposes that companies with sufficient technological resources offer software agents that, having been contracted as services, participate on behalf of clients in virtual environments. These software agents, called ***broker agents***, are responsible for complex behaviors, including the development of action plans, cooperating and competing with other agents, and conducting negotiations for the exchange of resources. For the client side, the AS model proposes the installation of one or more *light* agents in constant communication with the broker agent. These other agents, called ***local agents***, are responsible for simple tasks, such as applying commands (defined by the broker agent) on local resources and sending the broker agent information about the current state of the resources or about new directives defined by the user.

In essence, the AS model proposes a functional break-down of the tasks that a single agent usually tackles in a multi-agent system, so that the client's objectives are accomplished by a remote, broker agent in communication with one or more local agents. In order not to lose the simplicity that characterizes cloud computing, both types of agents are supplied by the same third-party provider located in the cloud. In that respect, it is particularly important that the installation of the local agents be automated in a way that requires no intervention of the user.

In line with cloud computing solutions, the AS model consists of the following main entities (Figure 3.1):

- *Agency Services Provider (ASP)*: A company with the knowledge and the technological infrastructure to deploy software agents that represent the interests of users in virtual societies.
- *Business Site*: An entity that creates virtual environments in which all the ASPs that fulfill specific rules and social conventions can participate through well-defined interaction mechanisms. Common business sites may be buying and selling sites on the Internet, computational grids and intelligent networks, including the Smart Grid.
- *Client*: A user or company that contracts the services of an ASP in order to participate in the virtual environments instantiated by a business site. As a result of its participation, the client expects to perform specific tasks or make profit. In principle, clients do not have the necessary knowledge or resources to develop their own solutions, or to do so profitably.

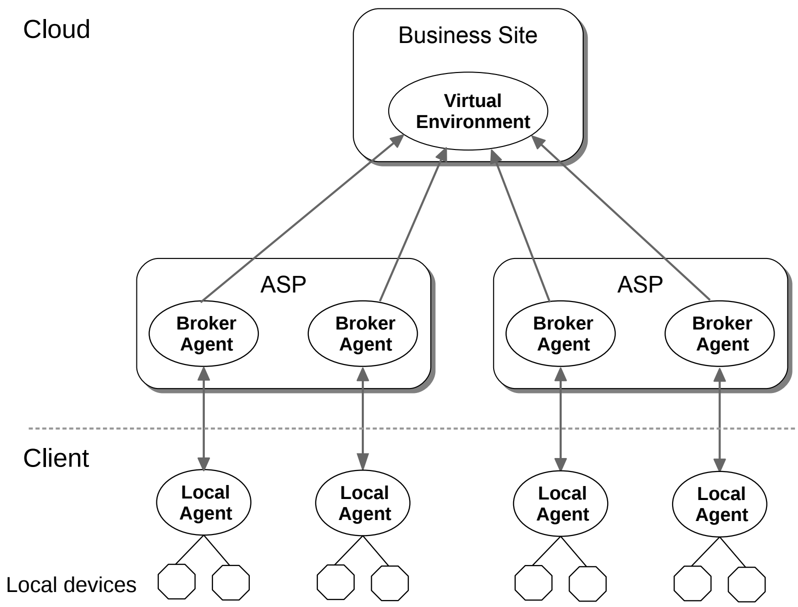

Figure 3.1: General scheme of the Agency Services model.

To shed more light on the purpose of the AS model, an example based on eBay is illustrated. In particular, eBay can define interfaces through which external software agents can bid on offers, and receive information about standing bids and deadlines. In this case, ASPs authorized by eBay can be contracted by users in order to effectively automate their participation in the auctions, thus saving them from having to develop their own solutions. ASPs also mean a guarantee for eBay, since they may impose a minimum set of rules and social conventions that ensure the proper operation of the site. This example, where the *business site* is eBay and the *virtual environments* are auctions, can be easily applied to more challenging technological fields where software agents are called in to play an important role, such as grid computing and the Smart Grid. In these cases, in addition, a role such that played by the local agent gains importance because it has to apply the actions determined by the broker agent on the local resources.

Note that nothing prevents the user from having more than one local agent assigned at the same time. Thus, a local agent can automatically inform the broker agent about the state of the resources, meanwhile the user may use another local agent installed in his/her mobile phone for both receiving information from the broker and, if necessary, sending it new directives. Therefore, the broker agent, apart from participating in virtual societies, can also work as a proxy agent able to communicate local agents with each other (Figure 3.2).

The main stages through which clients go in the AS model are:

- *Registration*: The client registers with an ASP that has been previously certified as reliable by the target business site. The main aspects of the agreement are the duration of the service, the processes in which the client wants to participate, and the configuration of the broker agent that will represent the client.
- *Configuration*: If necessary, the client installs local agents in its devices and configures the brokering service through software applications.
- *Expiration*: Once the contract expires, the ASP suspends the participation of the client in the virtual societies of the business site.

The user can set the preferences of the brokering service through web interfaces or mobile applications, both provided by the ASP. This information is forwarded to the broker agent when it is instantiated. Thus, in settings with local resources,

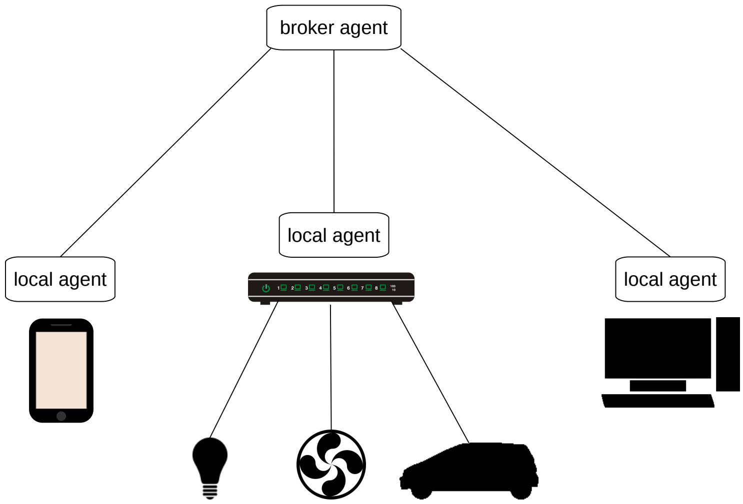

Figure 3.2: Broker agent working as proxy of multiple local agents.

there is no need for embedded interfaces in order that the user communicates with local agents and resources.

Once the client has contracted the services of an ASP, its participation consists of three main stages:

- *Initiation*: The business site informs the ASPs that a new business process has been initiated. Each ASP deploys a broker agent for each client. If necessary, the ASP automatically updates the local agents' software. The broker agent tells the local agents a new negotiation process has begun and, if any, the local agents report on the status of the local resources.
- *Execution*: Broker agent is registered with the virtual environment and, according to the state of resources and the user preferences, it interacts with other broker agents developing action plans, coordination tasks and negotiations. Throughout this process, the broker agent may periodically inform the local agents about its participation in the virtual environment. Moreover, when there are local resources, the broker agent transmits the actions to be applied on them. Also, the local agents inform the broker agent of new local events and new directives (defined by the user).
- *Close*: The broker agent records the details of its participation in the virtual society, and tells the local agents that the process has been completed.

It is possible to build a simpler version of the AS model. Specifically, the roles of business site and ASP can be joined in a single node (Figure 3.3), so that both the virtual environments and the broker agents are provided by the same entity. Although in doing so some level of competence is lost, this solution still maintains useful features, since the ASP can still offer users the possibility of configuring how the broker agent must behave. Continuing the example based on eBay, in this simplified version of the AS model, eBay may provide software agents that users contract in order to automate their participation in auctions. In this case, eBay would simultaneously work as business site and ASP. In order not to lose the autonomy and independence that intelligent agents are supposed to provide, eBay may allow users to configure the behavior of the software agents through directives. Furthermore, eBay may offer advanced behaviors in exchange for more expensive rates.

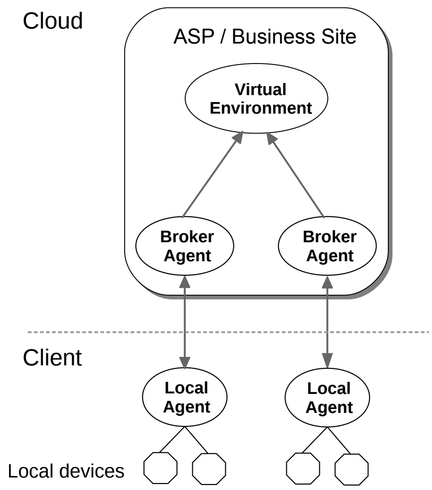

Figure 3.3: Simplified version of the Agency Services model.

3.3.2 Technologies

The aim of this section is to prove the technical feasibility of the proposal. To this end, actual technologies that address the major challenges of the AS model are presented. However, it is noted that using other technologies is also possible. The two main challenges that the AS model faces are the ability to: (*i*) conduct asynchronous, bidirectional dialogues between remote software agents; and (*ii*) dynamically deploy software agents in remote devices. Well-known technologies that successfully solve these challenges are:

- The *Extensible Messaging and Presence Protocol* (XMPP) for the communication between the broker and local agents. This is an instant messaging protocol [XMP] based on XML that supports secure communications. Although XMPP is usually associated with applications such as Jabber and Google Talk, it was actually designed for communication between agents, whether they are human or software. As a matter of fact, there are already solutions using XMPP to encapsulate and send FIPA messages [GPA02]. As for the infrastructure, XMPP needs an instant messaging server, which can be installed in the ASP infrastructure.
- *Java Network Launching Protocol* (JNLP, [Ora00]) for transferring and launching the local agent from the ASP. JNLP is a protocol for downloading and launching remote Java applications. It is a mature and widely used technology that ensures the latest available version of the software package is launched. Furthermore, it uses digital certificates to guarantee the authenticity and integrity of the application.

In general, the use of instant messaging protocols provides an easy and effective way to communicate remote software agents (as the interaction between broker and local agents requires), thus avoiding the need for using more complex mechanisms, such as those based on static IP addresses and web services. In particular, web services are not a feasible technology for this goal because it requires that one of the two nodes installs a web server, and does not provide asynchronous, bidirectional communications. On the other hand, JNLP ensures that the client can remotely install and launch local agents in a transparent manner, thus conserving the simplicity that characterizes the cloud computing model. Furthermore, both XMPP and JNLP consume few resources so that they can be used in embedded systems and modern devices, including mobile phones and tablets.

The communication mechanism for the interaction of broker agents with virtual societies is defined by the business site and is transparent to the customers, as it is an issue taken on by the ASP. The mechanism can be based on specific agent-based frameworks. In any case, this point does not represent a technological risk as both business sites and ASPs are supposed to be technological companies with sufficient knowledge and resources.

3.3.3 Benefits and virtues

One of the major goals of the AS model is to answer the challenges faced by intelligent agents in its quest to become a more accessible technology. In this regard, as described, when software agents (not just specific functions of them) are offered as cloud services:

i. The complexity that entails developing software agents able to participate in virtual environments is delegated to third-party services providers. In addition, the need for updating agents in order to improve their performance or to adapt them to both the interface and social conventions of the target virtual environment is responsibility of services providers.

ii. Users can contract and interact with agents using any device with Internet connection and thus participate in virtual societies without hindering mobility.

iii. Users can pay for specific capabilities, thus determining the scope and skills that broker agents can develop in virtual societies.

Revisiting the lessons learned from fields facing similar challenges (see Section 2.5), it can be noted that the AS model replicates many virtues of previous successful solutions. In particular:

- ASPs contribute to building *virtual breeding environments* (see Section 2.5.3, page 45). As commented, a business site evaluates the capabilities of the ASPs wanting to participate in future business opportunities. This condition ensures that all software agents deployed by the ASPs meet behavioral conditions and share both common ontologies and communication technologies. In addition, ASPs do not suffer from lack of participation, which is a restriction traditionally attributed to the solutions based on the VBE concept. On the contrary, the ASP role is designed to instantiate thousands of nodes, being all of them considered valid. Therefore, ASPs help to normalize the environment, while preserve the autonomy of the customers and promote their participation.

- As shown in Figure 3.1, solutions based on ASPs provide an architectonic structure very similar to that of hybrid P2P networks (Figure 2.10, page 42), which are characterized by the concept of supernode. In practice, ASPs are supernodes that principally develop advanced brokering functions on behalf of other nodes. In addition, they can provide other interesting behaviors such as accessing third-party services to obtain and process information that may be necessary for the broker agents and the local nodes.
- Many of the conclusions reached by the community devoted to grid computing are actually part of the agency services foundation. On one hand, the AS model is built on the principles of service orientation and standardized communications, which is precisely the approach adopted by the latest grid computing development frameworks in order to improve interoperability. On the other hand, the AS model aims to achieve a compromise between using advanced mechanisms based on intelligent agents and delegating the most complex part of this technology to specialized companies, as well as to conduct the process in controlled environments. This compromise shares many characteristics with the practical vision that has brought grid computing to achieve operational solutions (see Section 2.5.2, page 43).

Furthermore, in general, compared to traditional multi-agent systems, the AS model offers advantages in the following aspects:

- *Participation*: The transfer of the most complex tasks to the cloud, together with the simplicity offered to the clients, makes it easier to automate the customers' participation in modern virtual societies.
- *Scalability*: Focusing the most complex technologies in companies that are intended to be powerful and specialized, brings out advantages of economies of scale, so that the solution can grow with little effort.
- *Flexibility*: The client can participate in more than one type of virtual society with no need for additional efforts. Furthermore, the client can choose the provider that best fits its needs.
- *Reliability*: The model permits criteria to be established for the ASPs so that the broker agents' activity does not endanger the stability of the system due to selfish or anti-social behaviors.

- *Competitiveness*: The nature of interactions, clearly oriented to facilitate the implementation of competitive models, helps create exchanges based on market mechanisms.

3.3.4 In relation to intelligent agents theory

3.3.4.1 Intelligent agents and services-orientation

In the world of software, services-orientation practically means *web services*, which are pieces of business logic accessible via standard Internet protocols. Their aim is that remote clients can build robust and complex structures based on loosely coupled and heterogeneous functionalities. Although the technology has been widely accepted for client-server communications, its specification suffer from characteristics that limit Internet options. The most prominent are the need to know in advance the definition of services to invoke, the absence of semantic information and the use of non-persistent communications which are always based on the *request-response* pattern.

According to W3C specifications, software agents are a necessary component to articulate the web services infrastructure [Bea04]:"*software agents are the running programs that drive web service, both to implement and to access them as computational resources that act on behalf of a person or organisation*".

In line with this approach, agents have been proposed to be part of the business logic of web services with the aim of providing intelligence and reactivity to their behaviors, filtering requests and searching for sources of information [CL07]. Furthermore, mechanisms have been proposed in order for agents and services can interact with each other in a transparent manner [GC04]. Thus, agents may exploit functionalities offered by both other agents and services available in the context.

However, in practice the absence of semantic information has significantly limited the applicability of these lines of work. In response, the community has been working since 2001 to transform Internet into a semantic web by means of ontologies and adopting standards for the description of resources. These technologies aim to enable software agents to reason about properties and functionalities of web resources and services [SBLH06]. In this subject, agents are normally proposed for orchestrating services and searching those that satisfy the goals of the client.

As shown, efforts that relate intelligent agents and web services generally aim to improve the functionality and accessibility of services. However, there are no proposals in the opposite direction: assessing the success of web services solutions in order to overcome the problems which hinder the popularization of software agents. The Agency Services model responds to this novel vision: it uses service-orientation ideas to create a new model that facilitates user access to the intelligent agents technology.

3.3.4.2 **Agents as intermediaries**

In intelligent agents theory many efforts have been devoted to developing the concept of *intermediary agent*, also known as *middle agent*. The objective of this type of agent is to assist in communication tasks in order to facilitate exchanges between *requesters* and *providers*. Requesters are agents with objectives they want to be achieved by other agents, whereas providers are agents that fulfill objectives on behalf of other agents. The presence of middle agents is especially valuable in distributed, open environments, where they constitute a mechanism to overcome the heterogeneity between partners. Although there are several roles for middle agents, three of them are mainly recognized [KS01]:

- *Matchmaker*: The functionality of a matchmaker agent corresponds to that of the yellow pages. Providers register their skills in the matchmaker agent. Requesters consult it in order to identify those providers that are capable of fulfilling their objectives. If the activity of the matchmaker is successful, the requester and the provider then enter into a new dialogue.
- *Blackboard*: The blackboard agent registers petitions corresponding to tasks to be done. Specifically, requesters send their petitions to the blackboard, whereas providers ask this for petitions they can fulfill. In addition, the blackboard agent is commonly proposed to keep track of the requests and their respective answers so that other agents can easily extract information later.
- *Brokering*: The aim of the broker agents is act on behalf of the requesters. It negotiates the requesters' petitions with the providers, and finally conducts the results to the requester. Therefore, in the models dominated by a broker, there is no direct interaction between requesters and providers.

In the FIPA protocols there is one specially dedicated to the interactions mediated by broker agents [FIP02a]. In short, the agent that initiates the interaction (*initiator*) delegates the accomplishment of a task to a broker. After sending the request, the initiator plays no further part in the process. In general, the FIPA specification proposes neither client autonomy nor the need to relieve the client of the technical details.

The Agency Services model is clearly based on the brokering role. The novelty of the model is that it delegates this functionality to the cloud so that broker agents are hired and work as a cloud service. Of course, the proposal also details and solves all the issues that arise as a result of externalizing the brokering role.

3.4 Agency Services for the Smart Grid

3.4.1 ASPs for the Energy Management (ASPEMs)

The architecture proposed for the Smart Grid in the OASIS and NIST standards leaves the door open to the installation of nodes with different profiles. In the particular case of OpenADR, the first version of the standard [PAG$^+$09] considered: (*i*) *simple nodes*, which automatically apply the DR signals they receive; and (*ii*) *smart nodes*, which usually work as aggregators of simple nodes, being able to process and transform the received signals. In this regard, output signals of smart nodes are usually designed to achieve the same result as the input signals, but respecting internal conditions of the sub-section managed by the node, including preferences and requirements of customers. Despite their advanced behavior, smart nodes cannot be considered as powerful as supernodes in P2P networks or ASPs in the AS model. Therefore, given that the AS model inherits many advantages of the cloud computing paradigm and replicates the virtues of existent successful solutions for similar environments (such as P2P networks), it is advisable to study the benefits that may arise from installing nodes with comparable characteristics to those of ASPs in distributed energy networks.

In response to this opportunity, this research studies the installation of ***Agency Services Provider for the Energy Management*** (ASPEMs), which is introduced as a type of node capable of adding intelligent management behaviors and providing advanced data services to the electrical grid's customers. Like aggregators, ASPEMs are nodes that implement both interfaces VTN and VEN simultaneously. What highly distinguishes an ASPEM of an aggregator (or a smart node in the case of DR architectures) is the manner in which incoming events are processed: instead of redirecting the events directly to the leaf nodes, or distributing them according to predefined criteria, ASPEMs are intended to providing advanced functionalities, including the ability to instantiate energy markets. In this regard, depending on the autonomy of users to defend their interests, two approaches are possible:

- *Distributed mechanism*: The broker agents coordinate or negotiate between themselves the signals they will send to the customers. This approach allows the ASPEMs to run internal energy markets in which broker agents participate according to the preferences configured by the customers. As a result of negotiations, the input signal is translated into new sets of commands to be sent through the VTN interface to clients.
- *Centralized mechanism*: The ASPEM runs an intelligent algorithm that decides which signals must be sent to each customer. In making the decision, the clearing algorithm can also consider the users' preferences. Under this approach, direct interaction between broker agents may be unnecessary.

Certainly, the most novel approach is that based on distributed mechanisms, since it allows responding to the incoming signals using markets.

One of the main goals of the Agency Services model is to simplify the infrastructure of the client by delegating the complex and advanced behaviors to entities in the cloud. Applying this condition to the Smart Grid and the OpenADR standard means that end-nodes, which are typically users' facilities, may adopt the simplest profile of the standard (i.e., profile *2.0a*), but still enjoying part of the advantages of the most sophisticated ones (i.e., profiles *2.0b* and *2.0c*) thanks to the action of ASPEMs and broker agents. For instance, this capability would permit converting signals of type *delta* (those that specify the amount to be curtailed) into sets of signals of type *simple*, which are restricted to using the values: *normal*, *moderate*, *high* and *critical* (see Section 2.3.2, page 34). This conversion can be done so that

ASPEMs simply take into account the users preferences (in a centralized approach), or even can defend their own interests through the action of broker agents (in a decentralized approach).

Data services are essential to make an accurate conversion between profiles. Without information about the context, customers and broker agents cannot truly determine what actions are most appropriate and profitable for their interests. As a matter of fact, the architecture of the Smart Grid always appears complemented with data services provided by third parties, mainly including demand estimates, weather forecasts and energy prices (see Section 2.4.2, page 39). Furthermore, it is commonly assumed that these services are consumed and processed by end-nodes. However, in practice, this feature may turn out to be very demanding for AMIs that are intended to be installed massively, as well as for their maintenance. In the Agency Services model, this ability, which indeed is more typical of data centers, can be assumed by the ASPEMs so they can either provide these services or obtain them from third-parties. Later on, this information is used by the broker agents during the planning and bargaining processes.

Figure 3.4 depicts the interactions of an ASPEM node as described in this section. Through the VEN interface, ASPEMs receive messages corresponding to any profile of the OpenADR standard; while, through the VTN interface send exclusively messages of the profile *2.0a* to the users' facilities. In order to make this conversion, which may be carried out through energy markets, ASPEMs access data services.

In the scenarios foreseen for the Smart Grid, ASPEMs may group thousands, hundreds of thousands or even more nodes. For all of them, the ASPEM provides brokering and data services, while simplifying the customers' infrastructure. Therefore, they have to be considered as much more than aggregators, standing closer to the role of supernode of P2P networks.

3.4.2 Compatibility with the standards

The architecture of the Agency Services model is compatible with the EI standard (see Section 2.3.1, page 32). The following correspondence between entities from both contexts can be established:

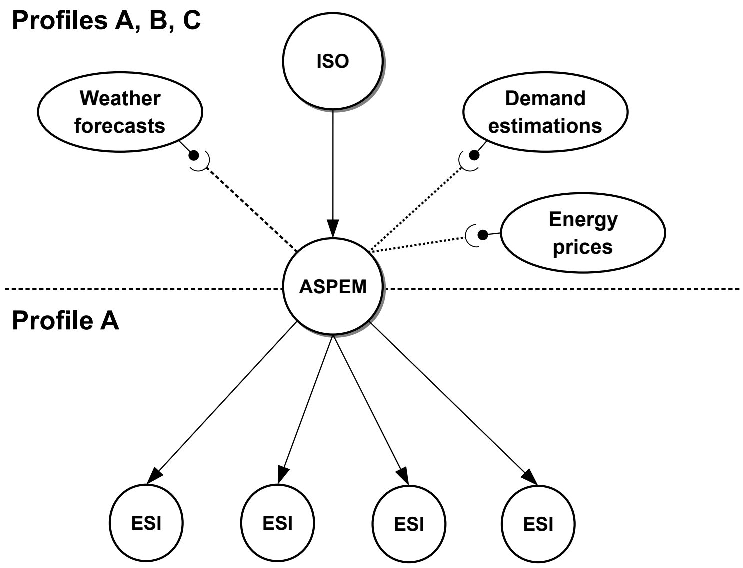

Figure 3.4: Interactions of an ASPEM node.

- ASPEMs are nodes that implement both interfaces VEN and VTN. In the original AS model, ASPs do not interact directly with each other. Instead, broker agents deployed by them interact in the target virtual environments. This condition conforms to both the EI and OpenADR standards, which do not allow peer-to-peer communications.
- The role of client in the AS model corresponds to a VEN node of the Smart Grid, which usually represents a user's facility. As with nodes of ASPs, in the AS model there is no direct communication between client nodes. Instead, interactions between them are carried out exclusively through broker agents in the virtual environment. If necessary, the local agent can be installed in the smart metering infrastructure, and its interface added to the ESI.
- The business site corresponds to a VTN node preferentially located at a high level of the infrastructure, owned by the system operator. In this case, the virtual environments are energy markets and DR programs, which are run through the technological infrastructure of the corresponding node.

Figure 3.5 illustrates the equivalence between both models. On the right side, the EI model is overridden with the roles that would participate in a Smart Grid infrastructure implemented in accordance with the Agency Services model. Note that, strictly speaking, ASPEMs do not have to be placed at the second level of the hierarchy. According to their definition, the conditions that actually impose the presence of ASPEMs in the architecture are:

i. In the tree of nodes, an ASPEM cannot be the parent of another ASPEM. The services offered by ASPEMs are meant to be contracted by end-users that want to automate their participation in energy markets or DR programs.

ii. The client nodes do not have to be end-nodes. However, an ASPEM only acts on behalf of its customer nodes, and not on behalf of the children it may have. Consequently, if an aggregator contracts the services of an ASPEM, the nodes managed by the aggregator are no explicitly represented by the ASPEM. In this case, the aggregator is responsible for wrapping the information and interests of the nodes behind it.

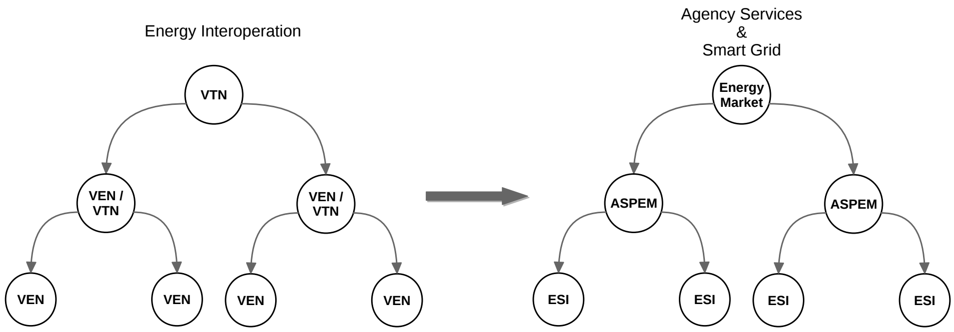

Figure 3.5: Correspondence between the architecture of the EI standard and the Agency Services model.

The capacity of upgrading the software of local agents (as stipulated in the AS model) is supported by the Smart Grid standards. Specifically, this task is addressed in the *Meter Upgradeability Standard* [PAP]. This action arises from the uncertainty surrounding Smart Grid; particularly from the need for utilities to ensure that existing technologies are interoperable and will comply in the future.

The OpenADR standard does not provide messages that allow the communication between ASPEMs and business sites located at different nodes. Consequently, only the simple approach of the Agency Services model is applicable in this case. Each ASPEM therefore must contain its own business site, so broker agents and virtual environments are instantiated in the same node. This condition implies that users are segmented into different markets that are held separately. However, despite having a large single market would certainly be more efficient, local markets still offer noticeable advantages over the traditional approach.

3.5 Energy markets in DR programs

With the aim of providing meaningful contributions, the work developed in this thesis is placed in the context of DR programs, which are one of the most awaited milestones of the Smart Grid. However, DR programs represent a challenging context for implementing distributed market-based management systems, since in this case consumers are simply expected to alter their consumption in response to signals sent by the system operator. Thus, in order for the AS model to be able to fully exploit its potential in DR programs, it is first necessary to create a conceptual overlay that introduces the roles of producer and consumer. This section is devoted to that goal.

In fact, under special conditions, some nodes play roles similar to those of the buyer and seller. These are:

- *Negative load*: A node, intentionally, consumes less than expected. In this behavior, the node can be considered a producer because it generates a deficit of demand that can be consumed by other nodes.
- *Critical load*: A node that, due to special circumstances, is guaranteed a minimum amount of power. In environments such as DR programs, where all nodes are supposed to limit their consumption in unison, nodes acting similarly to critical loads are consumers, since it is necessary to supply the surplus of demand they generate.

On the basis of the above concepts, in order that DR programs are able to host exchange processes, three roles are proposed:

- *Easy-load*: The node determines that it is willing to discard energy up to a specific consumption level during a time period. For instance, although a signal of moderate level may arrive, the node may set that it is willing to adopt the `high` or `critical` levels. This type of action is supposed to be done in exchange for a fee, although there may be customers willing to do so without compensation. In practice, easy-loads act as producers, because leading the consumption capacity to more restrictive levels is negative demand, which is equivalent to positive generation capacity.

- *Hard-load*: The node determines that it would like to protect a minimum consumption level during a time period. For instance, a node can define that it would like to avoid taking from the moderate level onwards. Hard-loads act as consumers because they strive to demand more energy than allowed.
- *Normal-load*: The node sets no special behavior, so it will apply the consumption level specified by the incoming signal during that specified time period.

The aim of the above roles is to build DR environments where hard-loads, in order to avoid applying specific DR signals, strive to purchase energy blocks from easy-loads, which are willing to discard more energy than required. It is important to note that a hard-load may need to acquire energy blocks from many easy-loads to cover all its demand, so negotiations are not actually limited to pairs of nodes. If a hard-load does not cover the amount of energy necessary to avoid adopting a specific level of consumption, then the node is required to apply that level. In this way, it is ensured that the amount of energy discarded is always equal to, or greater than, the amount defined in the incoming DR signal.

By using ASPEMs, the task of negotiating the exchange of energy blocks falls to the broker agents, which then convey the result to the local agents in form of profile *2.0a* signals. In this way, the requirement of the local devices is minimized. On the other hand, customers are only in charge of setting their preferences, including parameters such as:

- Time periods during which they want to participate as easy-, hard- and normal-loads.
- The price at which they are willing to purchase and sell energy at each of the time periods defined by the customer. In addition, the customer can state the maximum and minimum prices at which he/she is willing to purchase or sell energy blocks.

- Priority and additional features offered by the ASPEM. Customers can contract to the ASPEM the priority according to which they will be called when they act as hard- and easy-loads.

Since ASPEMs are technology companies, it is plausible that users can set their preferences through applications for mobile devices and websites. These means represent a great improvement over solutions based on interfaces typical of AMI devices, which are limited by definition. The configuration of the broker agents by using mechanisms typical of service-based solutions is another feature that the Agency Services model inherits from the cloud computing paradigm.

All the above concepts and features enable ASPEMs to create energy markets in rather limited environments such as the DR programs.

Escucha. Atiende. Vamos a realizar juntos el escrutinio de la escritura. Te enseñaré el difícil arte de la ciencia escriptural que no es, como crees, el arte de la floración de los rasgos sino de la desfloración de los signos.

"*Yo el Supremo*", Augusto Roa Bastos.

CHAPTER

4

Review of the agent-based algorithms for DENs management

The previous chapter introduced an architectonic solution aimed at facilitating the use of software agents in large distributed environments such as the Smart Grid. This chapter, in order to provide a complete solution, is devoted to identifying algorithmic solutions which use software agents for the management of DENs. The extensive literature devoted to solving similar problems leads us first to study the state-of-the-art, which gives the opportunity to harness valuable insight from existing proposals and, not least, contribute to their realization and improvement. However, at the present moment, there are no reviews dedicated to the assessment of agent-based algorithms for DENs management. The text of this chapter fills this gap. In particular, in order to fully exploit the capacity of the Agency Services model, the following study focuses on research that facilitates the implementation of the SDM model (see Section 2.2.2, page 28), so it is principally centered on works based on market mechanisms.

First, this chapter characterizes energy markets by describing all features and conditions that algorithms designed to implement efficient energy management systems must satisfy. Secondly, the text proceeds with the presentation and review of algorithms that, in principle, have been designed to implement management systems based on the SDM concept. With the aim of drawing useful conclusions, the algorithms are classified according to the properties required by energy markets. This classification serves as starting point for a discussion that will provide understanding of their suitability, advantages and weaknesses, paying particular attention to their completeness and efficiency. This review of the state-of-the-art ends up remarking on the most promising studies and pointing out the aspects that need to be reinforced in future research. In addition, as this work is closely related to artificial intelligence, this review includes a section on typical techniques used in this research field which are commonly proposed for covering ancillary tasks.

4.1 Characteristics of energy markets

In energy markets the exchanged good is electrical energy, and the participants are software agents that make offers for producing and consuming it on behalf of customers. Despite lessons learned in electronic commerce, energy networks have special features that make energy markets especially difficult to manage:

1. Supply and demand must be balanced continuously in order to ensure the proper operation of the network.
2. Energy is not a good that can be stored on a large scale, so all the energy that cannot be consumed at the moment has to be discarded.
3. Supply and demand depend on uncertain factors and therefore it is necessary to work with forecasts and estimates.
4. Reactivity of consumers and producers is limited and slow.

In addition, the need to ensure efficient use of resources and guarantee that users can satisfy their demand without excessive risk-taking, make that energy markets are rather difficult to implement. Specifically, fully functioning energy markets are:

- *Multi-unit*: Consumers and producers can negotiate a variable amount of energy, so more than one unit of the good can be exchanged. In this case, bids are usually expressed in form of linear piece-wise functions [DJ03, SS01] (Figure 4.1), in which price increases in relation to the amount; while the clearing price is determined by the amount of energy exchanged.

- *Multi-item*: Energy markets are fragmented into time slots. In practice, each slot is a market item that can be negotiated by the participants.
- *Combinatorial with complementary goods*: This refers to the support of bids that include groups of items that must be accepted or rejected as a whole. In the case of energy markets, it refers to the possibility of submitting bids that span multiple consecutive time slots.
- *Combinatorial with supplementary goods*: This refers to the possibility of making bids including several groups of items, so that only one of them can be accepted. In energy markets, this feature means that a customer can define multiple periods of time in which a specific amount of load or generation can be accepted.

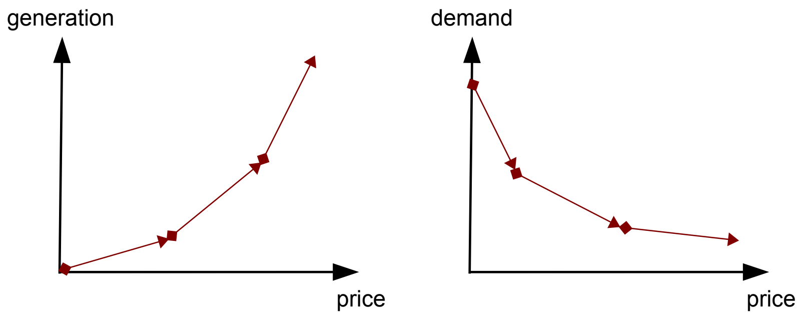

Figure 4.1: Linear piece-wise functions that define the behavior of the generation and consumption of energy.

Supporting combinatorial bids, despite the complexity that it entails, is an important requisite of energy markets. For instance, it is usually necessary that generation devices such as oil-based engines have to be active during a minimum period of time to be profitable and efficient. Likewise, some types of loads, such as washing machines and dishwashers, may need to span their activity during multiple consecutive time slots in order to finish their work. In these cases, if submitting bids for complementary goods is not supported, customers have to send separate bids for each time slot, which means the customer must risk that not all bids are not accepted or rejected jointly. On the other hand, bids of supplementary items allow that devices such as washing machines and water heaters can define disjoint sets of time slots in which their demand can be supplied, thus allowing the system to shift part of the demand at peak times, and giving flexibility to the providers. The importance of combinatorial bids, however, is not reflected in the state of the art, which usually defines energy markets without considering these properties.

Markets that meet all the above characteristics are very difficult to resolve. As a matter of fact, in medium to large sized environments, the search for an optimal solution is considered a NP-Hard problem [RPH98, San02]. As a result, in practice, solutions are based on simplified models, so that heuristics are used to accelerate the search for solutions, and, when possible, particularities of the case under study are exploited.

Furthermore, the complexity of energy markets is increased by the properties that are expected from any clearing algorithm that operates in the context of the Smart Grid, which should have:

- *Responsiveness*: In distributed energy environments, planning is short-term, using time horizons that can range from fifteen minutes to few hours, so the algorithm must be able to quickly find a solution.
- *Reliability*: The solutions proposed by the algorithm must be able to meet goals imposed by the SO, such as energy quality and supply reliability.
- *Scalability*: In contexts typical of the Smart Grid, the number of nodes can grow significantly, so it is important that the performance of the algorithm, as well as the quality of the solutions, scale successfully.
- *Autonomy*: The solution provided by the algorithm must strive to meet the preferences of users. The system goals may conflict with individual preferences, so the algorithm should also be able to find a workable compromise.
- *Reactivity*: In the electrical grid, planning is performed on the basis of forecasts and estimates, so unbalances between supply and demand is a frequent reality. Therefore, the clearing algorithm must be able to react to unexpected conditions.
- *Flexibility*: The algorithm must be able to adapt itself to unexpected events and eventual orders from the SO.

Despite all the advantages that the involvement of software agents promises, all the above-mentioned characteristics of energy markets, including the presence of combinatorial bids, make developing a fully effective solution a complicated task.

4.2 Algorithms

This section reviews representative works that describe solutions based on software agents for the management of distributed energy areas such as energy cells and micro-grids. Two main categories of solutions are considered: (*i*) those that propose micro-energy markets based on the SDM model; and (*ii*) those that are intended to providing ancillary services and manage emergencies. In both cases, the review is mainly devoted to techniques that come from the artificial intelligence field and electronic commerce.

4.2.1 Supply-demand matching

4.2.1.1 Double-sided auctions

In ordinary auctions, offers from all agents are collected by a central authority called the auctioneer, who is responsible for determining which are the winning offers and how the resources are distributed among them. When both producers and consumers can submit bids and offers, auctions are said to be *double-sided* or *two-sided auctions* [FR93]. Even though this approach seems to be suitable for solving most of the problems, in practice it is hard to implement an algorithm that, after evaluating all bids and offers, determines which resources are assigned to which entities. In this regard, the most basic implementation is to define a clearing price in which all offers that exceed it are accepted. This solution, despite its simplicity, has shown to be valid and speedy for environments that are not expected to grow beyond projected boundaries.

Focusing on the literature of the Smart Grid, *Dimeas and Hatziargyriou* (2004, [DH04]) uses English auctions to distribute generation resources between consumers in micro-grids. In this work, energy markets are held for 15-minute periods. Producers and consumers work, respectively, as auctioneers and bidders who compete for blocks of energy. To ensure competitive prices, a Grid Operator agent announces the prices at which the main grid is willing to sell and purchase energy. Furthermore, the time limit for negotiations is 3 minutes. *Ramachandran et al.* (2011, [RSEC11]) enriches this work by using a mechanism that minimizes the fuel cost on the generation side, and also implements strategies for handling the players' risk attitude at the trading period. Specifically, in order to optimize the generation costs, the authors have developed a novel algorithm that combines techniques from *Artificial Immune Systems* (AIS) and *Particle Swarm Optimization* (PSO).

Both proposals provide detailed experiments that simulate the operation of small micro-grids. In both cases, the participation of software agents that exchange energy blocks by using double-sided auctions has proven to bring benefits. However, it is important to note that none of the algorithms supports the negotiation of multiple items at the same time: bids and offers are always intended to cover the period ahead. Therefore, for larger and more complex scenarios of the Smart Grid, it is necessary to enable the algorithms to handle multi-item and combinatorial bids. Moreover, it is also necessary to point out that, in general, double-sided auctions scale very poorly when these conditions are considered, as is the case when the number of nodes grows, rapidly resulting in NP-Hard problems.

4.2.1.2 **Parallel auctions**

When the complexity is too high, parallel auctions offer an attractive alternative. In this scheme, each seller has the option of holding its own auction, so many auctions may be running simultaneously. Although sellers may accept combinatorial bids, they usually occur in the context of single-sided auctions that are significantly simpler and faster than the double-sided option. On the other hand, the main drawbacks of parallel auctions are:

- Using distributed local clearing algorithms to obtain the solution causes loss of global insight and, consequently, the capacity to obtain optimal solutions.
- Requiring the software agents to hold and manage their own auctions may be a demanding feature that can affect the level of participation.
- Sending the same bid to more than one auction implies that the bidder is actually overbooking his/her capacity, which may lead to solutions that, in practice, cannot be implemented. Thus, overbooking entails an important risk for the

security and reliability of the electrical grid. On the other hand, when overbooking is forbidden, a specific block of energy can only be offered to a single auction, with the result that many of those blocks can remain unassigned at the end of the process, thus leading to a noticeable waste of scarce resources.

Despite these disadvantages, parallel auctions may be explicitly requested by the players to gain autonomy and control over the decision process.

Amin and Ballard (2000, [AB00]) presents a proof of concept based on parallel auctions. However, the auctions used in the study do not support multi-unit and multi-item bids. On the other hand, *Penya and Jennings* (2005, [PJ05]) introduces the algorithm ***mPJ***, which supports bids that are multi-unit, multi-item and combinatorial. The main characteristics of *mPJ* are:

- In order that generation follows demand (demand-driven-supply), as electrical networks require, the market is based on reverse auctions. That is, auctions are held by consumers, while producers submit bids for selling generation capacity. Therefore, contrary to the classical approach, the entity that wants to acquire the good holds the auction.
- Buyers make bids through linear piece-wise functions [SS01, DJ03] like the one depicted in Figure 4.1.
- Auctions are non-iterative, so consumers have to determine the winner after the first round of bids.
- The implementation is developed according to the Vickrey formula [Vic61], so producers are encouraged to value energy according to their real needs.

mPJ is a brute force algorithm that shows good performance when combinatorial bids are omitted. In the other case, due to the large number of combinations that may be involved, the performance of the algorithm drops noticeably. However, as shown by the authors, in real energy markets only part of the whole spectrum of possible combinatorial bids is useful. In practice, agents will be preferentially interested in packages of consecutive items since they allow conducting consumption and generation actions that span multiple time slots without taking risks. By limiting the set of possible combinations to the sets of items that are principally needed, the performance of the algorithm remains good and users do not lose significant action capacity. Despite *mPJ* being one of the most promising mechanisms for implementing energy markets based on software agents, its performance has only been tested through theoretical tests that are not focused on distributed energy environments, but simply on proving its capacity to scale. Therefore, it is still necessary to simulate *mPJ* in scenarios typical of the Smart Grid.

4.2.1.3 **Price-oriented search of equilibrium**

This is the most common approach in classical markets. Parties express their preferences through price functions, so energy price is the factor that determines the amount of energy that each party will consume and produce. In this approach, the goal of the clearing algorithm is to find the price that optimizes the resources assignment. The main drawback of searching in the space of possible prices is that the optimal solution cannot be obtained analytically, which makes the process time-consuming, using large amounts of computational resources. As a result, to tackle it, many works use the price resulting from matching the offer and demand aggregated functions, which is commonly known as *equilibrium price*, as it is supposed to result in the quantity of supply being equal to the quantity of demand. Furthermore, if the market needs to be corrected, this approach allows that the authority can change resources distribution by simply altering price signals.

Arnheiter (2000, [Arn00]) describes a solution that uses price-oriented search of equilibrium to manage energy systems. However, this preliminary work does not support multi-unit and multi-item bids. Moreover, agents cannot use utility functions. *Logenthiran et al.* (2008, [LSW08]) implements a pool market that operates in the same manner as typical wholesale energy markets. The central agent represents the pool, which, after receiving all buying bids and selling offers from loads and generators, is in charge of determining the clearing price. This price corresponds to the highest accepted selling (generation) offer. As in the wholesale pool markets, all accepted generation bids are paid the clearing price, while loads are required to pay at that price. In general, pool markets are not intended to cover the features described in Section 4.1.

4.2.1.4 **Resource-oriented search of equilibrium**

The optimal solution can also be sought in the space of possible resource allocations. In this case, the algorithm searches for the allocation of resources that yields the equilibrium price. Specifically, it is said that the system is in equilibrium if, after all resources have been assigned, all agents are willing to pay the same price for a specific good. The system is therefore in equilibrium if it finds a Pareto optimal distribution of the resources. Compared with the price-oriented approach, this model has the advantage that agents' demand can be obtained analytically from utility functions, which significantly accelerates the searching process.

Ygge and Akkermans (1996, [YA96]) uses an algorithm based on this approach in order to manage distributed loads. The algorithm does not support multi-item bids, so the solution is not suitable for implementing environments driven by supply-demand matching. This problem is overcome in *Ygge and Akkermans* (2000, [YA00]), where the authors present the algorithm *COTREE*. This describes an environment where software agents send their utility functions to a central node that uses Newton-Raphson to find a Pareto optimal solution. *COTREE* assumes an architecture founded on hierarchical cells [KWK05], which, although standards are not explicitly considered in the work, is compatible with the architecture proposed by the OASIS Energy Interoperation standard. *COTREE* was implemented and tested as part of the *CRISP* EU project [ECN06]. The tests show that *COTREE* is fast and scales well, being able to find an equilibrium solution in less than one second for scenarios with hundreds of nodes. Also, simulations show that *COTREE* is effective for smoothing demand curves. However, the algorithm has the following handicaps:

- Combinatorial auctions are not supported. Accordingly, nodes cannot submit bids that span more than one time slot, which is a significant restriction for energy units.
- Any change in the agents' plans requires restarting the process, since a new equilibrium solution is necessary. It must be noted that the restoration involves all the nodes.

In general, the latter drawback is attributable to all solutions based on equilibrium search, since the solution that clears the market is calculated from the utility functions of all nodes, and therefore affects all of them. As a result, at the instant when a node changes its plans or cannot commit to fulfilling them, the equilibrium value is no longer representative, so a new global solution has to be calculated.

Carlsson and Andersson (2007, [CA07]) proposes an energy market based on a binary tree structure. A leaf node represents a market's time slot, while an internal node represents the union of the child nodes covered by it (Figure 4.2). The work is complemented with the algorithm ***CONSEC***, which supports multi-item and multi-unit bids. Agents can submit bids for any node of the tree, so bids corresponding to internal nodes are combinatorial. In particular, when using combinatorial bids, agents can determine whether the bid must be distributed among the child nodes (complementary bid) or assigned to the child that maximizes the interests of the customer (supplementary bid). Bids are defined through piece-wise linear functions (Figure 4.1). *CONSEC* takes advantage of the interdependence that actually exists between consecutive items in energy markets: when a node consumes or produces during a specific time slot, it will likely need to act on the next items. This circumstance usually converts the energy markets into a structure like the one depicted in Figure 4.2.

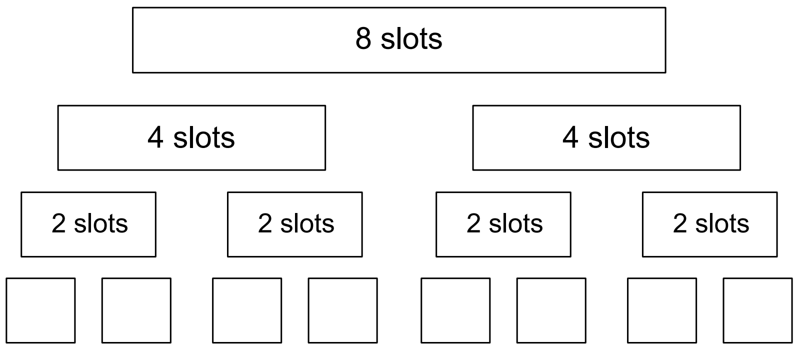

Figure 4.2: Time slots of energy markets expressed as a binary tree.

Once all offers and bids have been collected, *CONSEC* proceeds in two stages:

1. It conducts a resource-oriented search which aims to find the equilibrium price for each of the items of the market (for each time slot).
2. It assigns the bids containing supplementary items so that they are as profitable as possible to the involved nodes.

From both pieces of information (the equilibrium price and the set of supplementary items), nodes are able to know how much energy they have to produce or consume for each of the items to which they submitted bids.

As part of the experiments carried out within the CRISP EU project, *CONSEC* was tested in simulated scenarios that included a strong presence of distributed energy units [CRI02]. The results demonstrated that the algorithm scales well and succeeds in reducing the deviation between consumption and generation. In addition, the possibility of building combinatorial offers proves to be an advantage for the co-generation units, as well as for the programmable consumption units, which can plan their activities for multiple time slots.

However, the algorithm has the following handicaps:

- Bids involving disjoint elements cannot be sent, so supplementary bids are only supported to some extent: the involved nodes must be always grouped as child nodes at the same level. As a result, customers cannot submit offers for different sets of nodes with the aim of choosing the most profitable later. For instance, customers with flexible generation capacity such as hydropower plants or batteries can only send offers for a specific group of hours, and not for disjoint groups of them.
- An agent cannot include a specific generating unit or load within more than one bid, so the software agent must select the most valuable time slot for its interests.

Furthermore, *CONSEC* also shares the disadvantages typical of algorithms that use centralized searches of equilibrium:

- Users must express their intentions and preferences through utility functions. In complex environments such as those of distributed energy, building effective utility functions is a complex task that is usually addressed by using simplified models.
- It is assumed that the excess of demand and generation is taken over by the main grid or an external entity with infinite resources. Consequently, this type of algorithm is not valid for isolated environments or environments that are not allowed to generate demand or production beyond a limit.

- Behaviors of software agents can only be managed through price signals. This information is insufficient for defining priorities and other users' preferences.
- Any change in agent's plans requires that the entire process must be restarted in order to re-balance the system.

In general, the use of signals based only on prices, although they can be handled by intelligent agents, limits the algorithm, which requires additional mechanisms to deal with more elaborated strategies.

4.2.1.5 **Symmetric assignment**

In general, a symmetric assignment algorithm is formulated through *Persons* (P) and *Objects* (O), so that matching a person P_i to an object O_j has associated a benefit a_{ij}. Each object has a price p, which is determined from the bids that aim the object. The valuation of an object is therefore directly related to the interest of persons in acquiring it. On the other hand, the value of an object for a person is defined as the difference between the benefit that the object gives to the person minus the market price of the object (a_{ij}–p_j). The clearing process consists of an iterative algorithm that each time assigns an object to the person that obtains more revenue from it. The challenge of symmetric assignment algorithms is to find solutions that converge and obtain optimal solutions.

Dimeas and Hatziargyriou (2005, [DH05]) studies energy markets from the point of view of symmetric assignment problems. The authors design a mechanism that matches generation blocks to consumption blocks on a one-to-one basis so that the overall benefit is maximized. The problem is formulated so that generation blocks are persons, while demand blocks are objects. Since symmetric assignment algorithms require that the number of persons and objects is the same, the authors include a Grid Agent who is supposed to have capacity to provide the generation and consumption blocks that may be needed to balance the system. Also, in order to increase users' ability to purchase and sell energy, the Grid Agent doubles the initial number blocks with demand and generation blocks that come from the main grid.

Funabashi et al. (2008, [FTNY08]) develops a solution completely based on software agents so that, at market level, each block is represented by a software agent that is specially instantiated to that end. In practice, these agents work as brokers of the micro-grid central agent. As occurs in [Dimeas05], in order to match the number of load and generation blocks, if necessary, system agents and their corresponding blocks are created dynamically. In *Nunna and Doolla* (2013, [KND13]), the authors also propose a solution entirely based on software agents whereby micro-grids participate as individual nodes that strive to balance their internal generation capacity and load requirement.

The solution presented in [DH05] is tested on a laboratory micro-grid. The authors point out that, when the number of blocks to be matched is higher than 30, the performance of the algorithms falls, thus resulting that the process can take many hours to find the optimal solution. With the aim of reducing the number of blocks, the authors propose increasing the block's size. However, this approach affects the efficiency of the solution, as well as the volume of energy that agents can negotiate individually. Furthermore, negotiations only concern the period ahead, so managing multiple items is not possible. Due to these handicaps, symmetric assignment algorithms are unsuitable for managing medium to large sized energy environments.

4.2.2 Ancillary services and emergencies

4.2.2.1 Simple Contract-Net

As a way to give autonomy to users in small distributed energy environments, many works propose to use *Simple Contract-Net* [FIP03b], which is an interaction protocol widely used in agent-based solutions. The most known virtues of Simple Contract Net are that it is easy to install, and that it delegates the responsibility for processing the request to the local agents. However, it is also known that, due precisely to the simplicity of the protocol, its ability to express and negotiate deals is limited. In addition, the node that launches the request, after receiving all proposals, has to decide which of them are accepted, so the clearing mechanism is centralized. This condition may pose an important workload and complexity in some scenarios. As a result of these handicaps, in the context of the Smart Grid, Simple Contract-Net is only suitable for managing specific situations in small to medium sized environments. As a matter of fact, the protocol is commonly proposed to negotiations typical of ancillary services. Following this approach, *Lum et al.* (2005, [LKG05]) and *Jiang* (2006, [Jia06]) use Simple Contract-Net to negotiate the availability of batteries and other energy sources in small micro-grids. *Kim and Kinoshita* (2009, [KK09]) uses the Contract Net protocol for gathering generation bids that, in a later stage, are used to cover the aggregate demand of a small micro-grid for a specific period of time.

4.2.2.2 **Matchmaker**

In artificial intelligence, the concept of *matchmaker* [KS01] is used to describe a software agent that brings agents that require some work done into contact with agents that are capable of carrying out that work. To this end, the matchmaker registers and classifies services and abilities provided by agents called *providers*; and sends that information to agents who need them, which are known as *requesters*. When the action of the matchmaker is successful, the provider and requester enter into a new dialogue. This is why it is commonly said that the function of a matchmaker agent is comparable to that of the Yellow Pages.

Rahman et al. (2007, [RPT07]) uses this technique to build a management system for micro-grids whereby demand follows available generation (*supply-driven-demand*). Specifically, producers register their generation capacity with a central entity that sends the information to active consumers. Subsequently, consumers interested in a specific offer make contact with the selected producers. In this manner, consumers modulate the demand of their loads according to the volume of energy traded. Although authors do not mention the concept of matchmaker, the solution implicitly follows this approach.

In general, solutions based on the matchmaker role suffer drawbacks of centralized approaches, such as difficulty in scaling and a high level of vulnerability. However, as proposed in [RPT07], this type of solution can be especially useful for small environments such as micro-grids, as well as for handling business opportunities that may arise from applying ancillary services.

4.2.2.3 **Monotonic concession protocol**

The monotonic concession protocol [RZ94] consists of an iterative negotiation process between two agents. Its most significant feature is that, in order to ensure convergence, both agents are required to improve their offers at each iteration. This idea is used by *Brazier et al.* (1998, [BCG+98]) in order to build a load shedding mechanism. Specifically, the agent representing the utility, in anticipation of a peak demand, orders the agents to submit shedding offers iteratively until the system reaches equilibrium. In order to speed up the convergence of the process, the authors use an *announce reward table*, which is a structure that relates prices to amounts of energy. At each iteration, the table is populated with the set of offers that the utility agent is willing to accept. Therefore, consumers can only submit offers based on entries of the table. One of the key points of the work is that the utility agent controls the convergence of the negotiation process by modifying intelligently the content of the table, thus ensuring that a solution is achieved within a time limit.

Brazier et al. is not intended to support the SDM scheme, but following the classical model of the electrical grid, in which all consumers negotiate with one large source of generation. In particular, the solution aims to assist during moments of peak demand, as well as work as an ancillary service that adds value to consumers. Its main weakness is that the autonomy of consumers is limited, since they are forced to base their behavior on preset choices.

4.2.2.4 **Rules-based system**

Pipattanasomporn et al. (2009, [PFR09]) describes a solution in which agents act according to predefined sets of rules. The work is intended to support emergencies in which some agents strive to protect critical loads through programmed behaviors. In principle, due to their inherent limitations, the solutions based on rules are suitable for highly reactive, cooperative contexts such as single-facility micro-grids [YWM+05]. The main handicap of the rules-based systems is that they have no capacity to react to unexpected situations, so all conditions and states that may pose risk to the system must be considered in the design stage.

4.3 Discussion

The purpose of this section is to identify the most promising mechanisms, compare them, discuss their advantages, and propose new lines of research. To this end, studies are classified according to the properties required by the energy markets. In addition, information is given about how the studies have been tested, and whether it is possible or not to reproduce their experiments. Table 4.1 describes the characteristics used to classify the works, and Tables 4.2 and 4.3 show them classified according to different sets of those characteristics.

Table 4.2 shows that most works overlook all the challenges that distributed energy markets are expected to face. In particular, most of the algorithms do not support or do not even consider combinatorial bids. As explained, these types of bids are essential in order that customers can plan actions that cover multiple time slots, which is a usual requirement in energy contexts. As a matter of fact, only [CA07] (*Carlsson and Andersson*, 2007) and [PJ05] (*Penya and Jennings*, 2005) support this property in its two facets, including packages of complementary and supplementary bids. However, these two works are based on rather different approaches: the algorithm *CONSEC* presented in [CA07] looks for the equilibrium price by using a centralized scheme; while the algorithm *mPJ* described in [PJ05] proposes that each consumer holds its own auction, thus leading to a complete distributed solution based on software agents. The mechanism described in [CA07] suffers from drawbacks typical of centralized strategies and those based on equilibrium search (see Section 4.2.1.4). Furthermore, *mPJ* is more flexible than *CONSEC* in building combinatorial bids, as these can be made up of disjoint items.

Table 4.2 also shows that auctions are the method most often considered for implementing agent-based energy markets. The reason is twofold: (*i*) auctions are flexible enough to meet all requirements of energy markets; and (*ii*) auctions is a typical approach of the community devoted to developing multi-agent systems because it is distributed and ensures the autonomy of agents in decision-making. However, theory says that obtaining an optimal solution by using bilateral auctions becomes a NP-Hard problem in large environments, so parallel auctions is, in principle, the only valid method for accomplishing this task. Therefore, in order to check the suitability of the solutions proposed in [DH04] (*Dimeas and Hatziargyriou*, 2004) and [AB00] (*Amin and Ballard*, 2000), it is still necessary to conduct simulations with realistic scenarios considering all requirements. For its part, the solution based on parallel auctions proposed in [PJ05] has still to be simulated in scenarios typical of the Smart Grid. In this particular case, besides checking the efficiency of the algorithm *mPJ* in a realistic context, it is also necessary to study how typical drawbacks of parallel auctions may affect the capacity of the algorithm to balance the network. In particular, the best-known drawbacks of parallel auctions are: (*i*) poor distribution of buyers to the sellers [Hop08]; and (*ii*) the risk of buyers overbooking their capacity or, on the contrary, the business opportunities they may lose when they adopt overly conservative strategies.

In practice, the only works that have been simulated by using an electrical grid simulator are the ones carried out in the CRISP EU project. This is because defining realistic scenarios and simulating them is a complex task. Furthermore, the integration of intelligent agents in the Smart Grid requires a multidisciplinary team with specialized members in both areas. As a result, it is normal that works coming from teams principally devoted to the research of software agents overlook facets such as simulations supported by tools belonging to the electrical engineering. For instance, simulators such as GridLAB-D, which is open source and agents-based (thus representing an excellent opportunity) is not used in any of the previous works.

On the other hand, Table 4.3 shows information corresponding to other interesting aspects. Among them is standard-orientation, which is rather limited in works that combine software agents with techniques typical of the Smart Grid. The main reason for this is due to the development of stable standards in this field, as well as the acceptance of them as such, has been late. This fact has contributed to create a distorted idea of the Smart Grid and its goals, with the result that authors are occasionally unable to precise the context to which their works are intended. In the quest for implementing energy markets in the Smart Grid, well-established standards are definitely a valuable source of information that provides authors the insight necessary to identify the areas in which they can contribute on a safe foundation.

In scientific and technological areas, it is important that people other than the authors can reproduce research and academic progress. Reproducibility ensures that results can be corroborated, and that other authors can conduct new research based on previous works, thus making progress faster and more reliable. The security, reliability and efficiency required for any solution that involves changes in the electrical grid, as well as the presence of cutting-edge technologies, makes reproducible research especially important for the Smart Grid, where any solution must be previously studied and tested in detail. The definition of frameworks intended to guarantee works reproducibility is increasingly common [Sto09, FC09]. However, none of the works reviewed in this document successfully meets the conditions of these frameworks. On the contrary, they ignored this aspect. However, the performance and efficiency of some methods presented in the state of the art could be validated due to the extensive knowledge we have of some mechanisms, as well as due to the availability of the resources involved. From this less strict point of view, as shown in Table 4.3, works based on auctions can be considered reproducible, since they use algorithms that are easy to implement, and their efficacy for the particular context they have been designed can be checked. In contrast, non-trivial formulas and factors come into play in works based on equilibrium search, so additional information is required in order to consider their experiments reproducible.

To summarize, *Penya and Jennings* (2005) is the most promising and complete approach to fully implementing functioning energy markets based on autonomous software agents. *Carlsson and Andersson* (2007) meets all the requirements energy markets require, but from a more centralized and deterministic approach. Therefore, when the goal is implementing markets based on distributed decision-making, *Penya and Jennings* (2005) is definitely more suitable. However, as mentioned before, this method must overcome the complications of parallel auctions, including the need for software agents which can conduct their own auctions. In this regard, adopting the Agency Services model could be very useful, as the responsibility for holding parallel auctions would fall on broker agents. It therefore seems appropriate to assess the potential of this combination. Furthermore, as already discussed, it is still necessary to check the performance of the algorithm *mPJ* using an electrical grid simulator and realistic scenarios.

Property	Description	Values
Multi-unit	Indicates if it supports multi-unit bids.	yes, no, — (not considered)
Multi-item	Indicates if it supports multi-item bids.	yes, no, —
Combinatorial complementary	Indicates if it supports combinatorial bids with complementary items.	yes, no, —
Combinatorial supplementary	Indicates if it supports combinatorial bids with supplementary items.	yes, no, —
Test type	Indicates the type of environment in which the proposal have	*sim*: Simulated using an electrical grid simulator. *test*: Tests the efficacy and scalability of the proposal without using simulators. *exp*: Experiment in a laboratory environment.
Reproducible	Indicates if the proposal is reproducible.	yes, no
Standards-based	Indicates if it is based on the Smart Grid standards.	yes, no

Table 4.1: Properties for describing the works that aim to implement the SDM exchange model at medium to large DENs.

Work	Multi unit	Multi item	Comb. Comp.	Comb. Supp.	Test	Type
Dimeas et al. (2005)	yes	no	no	no	exp	Symmetric assignment
Funabashi et al. (2008)	yes	no	no	no	sim	Symmetric assignment
Nunna and Doolla (2013)	yes	no	no	no	sim	Symmetric assignment
Dimeas et al. (2004)	yes	—	—	—	test	Double sided auctions
Ramachandran et al. (2011)	yes	—	—	—	sim	Double sided auctions
Arnheiter (2000)	yes	no	no	no	test	Equilibrium
Logenthiran (2008)	yes	no	no	no	test	Equilibrium
Ygge and Akkermans (1996)	yes	no	no	no	test	Equilibrium
Ygge and Akkermans (2000)	yes	yes	no	no	sim	Equilibrium
Carlsson and Andersson (2007)	yes	yes	yes	yes	sim	Equilibrium
Amin and Ballard (2000)	yes	yes	—	—	test	Parallel auctions
Penya and Jennings (2005)	yes	yes	yes	yes	test	Parallel auctions
Rahman et al. (2007)	yes	no	—	—	sim.	Matchmaker

Table 4.2: Description of the works according to the characteristics of energy markets.

Work	Reproducible	Standards-based
Dimeas et al. (2005)	yes	no
Funabashi et al. (2008)	yes	no
Nunna and Doolla (2013)	yes	no
Dimeas et al. (2004)	yes	no
Ramachandran et al. (2011)	yes	no
Arnheiter (2000)	yes	no
Logenthiran (2008)	yes	no
Ygge and Akkermans (1996)	no	no
Ygge and Akkermans (2000)	no	no
Carlsson and Andersson (2007)	no	no
Amin and Ballard (2000)	yes	no
Penya and Jennings (2005)	yes	no
Rahman et al. (2007)	yes	no

Table 4.3: Description of the works according to their standard-orientation and the capacity to reproduce experiments based on them.

Hay momentos para recitar poesía y momentos para boxear.

"*Los detectives salvajes*", Roberto Bolaño.

CHAPTER

5

Simulation infrastructure

This chapter introduces the simulation infrastructure used to evaluate the theoretical research done throughout this thesis. Its design is marked by the need for simulating both the electrical grid and multi-agent systems, which leads to an infrastructure composed of multiple modules. The following sections describe the technologies used to implement the infrastructure, the purpose and main characteristics of each component, the interactions between them, and the life cycle of the simulation process.

As will be seen, the choice of technologies is guided by principles which are part of the aim of this project. In particular, the infrastructure is designed to comply with standard-oriented solutions based in extensively proven tools, and respects and promotes the conditions of reproducible research. Although it might seem that some of these conditions could limit the scope of the present project, the reader will have the opportunity to see that the best and most capable simulation tools in both domains (the electrical grid and multi-agent systems) have been implemented according to this thinking, with the result that, in practice, the main challenge relies heavily on achieving an effective liaison between simulators.

5.1 Simulation software

The research presented in this document seeks to be reproducible and verifiable by the research community. As described in reproducible research frameworks [Sto09, FC09], conditions of both the software and data used during the experimental evaluation play essential roles in achieving this goal. In particular, frameworks stress the need to meet the following features:

- Both, software and data sets must be accessible. In the case of software, it must be possible for users to obtain all software components required to perform the simulations. Undoubtedly, this need is facilitated when the software does not cost anything and can be obtained directly from Internet. As for data sets, there must be a description of how the data was brought into the form used in the research. In this regard, widely accepted data sets are considered valuable resources.
- The licenses for both must not impose conditions limiting the reproduction of experiments and dissemination of results.
- The software must contain complete instructions on how to execute and use it, as well as information about how to obtain and use the data sets.

On the other hand, the activity of the electrical grid implies the participation of many subsystems, which, in practice, results in a multilayer architecture that enters multiple domains, ranging from those focused on the electronic behavior of base components, to those focused on tasks related to the long-term management of generation and consumption resources. This condition makes it particularly difficult to capture all the complexity of the electrical grid with a single piece of software. As a result, simulation tools cover only some specific aspects of it; power flow calculations and demand models being the most attended functionalities. For this reason, research projects often resort to ***co-simulation*** [LSS+11, GMD+10, LAH11, LXJM12], a term used to refer to the need to simulate and model coupled problems in a distributed manner, so that subsystems are simulated separately, interacting with each other by using communication channels. Thus, in simulating the electrical grid, projects dealing with domains other than the two mentioned before are commonly required to use additional simulation frameworks. This is the case, for instance, of the project presented in this dissertation, which proposes a new architectural solution and the implementation of agent-based markets. Therefore, to make co-simulation possible, it is important that the electrical grid simulator provides its functionality through conventional means (such as APIs or web services), or, alternatively, facilitates the development of plug-ins that can accomplish this task.

Given the large number of components and subsystems that participate in the operation of the electrical grid, in order to ensure the experiments' reliability, it is also important that the simulator has been tested against benchmark scenarios, such as the *Distribution Test Feeders* [Chr99] defined by *IEEE Power and Energy Society*. Additionally, this feature improves the reproducibility of the projects tested on the simulator.

A description of the characteristics and qualities of the most capable simulators of the electrical grid can be found in [LZ14, RLS+14, Ste14, FCD+13]. For the present project, the **GridLAB-D** simulator [CSG08] was chosen to perform the experimental evaluation because, besides being one of the most promising options, it meets all the requirements defined above. In particular, GridLAB-D has been developed by the US *Department of Energy* (DOE) as a tool for facing the forthcoming challenges in the energy field. Some interesting characteristics of GridLAB-D are:

a) It follows an agent-based approach and is extensible, with modules which can simulate a large variety of components of the electrical grid at different levels of abstraction, including the power flow model and other physical constraints.

b) It is open source, with a growing, active community of both developers and researches working on it. In particular, GridLAB-D uses the *Berkeley Software Distribution* (BSD) license, which guarantees that users can make use of it in all possible manners, and that reproducible research principles can be applied to any derivative work.

c) Its efficacy has been extensively tested. It is worth mentioning that GridLAB-D includes the definition of the standard scenarios defined in [Chr99], for which it yields the expected values.

d) It is well documented, the information being updated regularly. In addition, it has active forums.

e) It is cross-platform with support for the most common operating systems, including *GNU Linux*.

f) It is designed to be extended through an advanced plug-in system that allows access to the simulator's kernel services.

As for software agents, there are many simulation tools. Reviews and comparisons of their characteristics can be found in [KB15, All10, BBD+06]. Among all of these, the framework most frequently used by the academic and industrial community is **Jade** [BBCP05]. This is an open source framework for implementing distributed agent-based systems, and stands out for being compliant with the FIPA specifications [FIP96]. Jade has been developed in the Java programming language and, like GridLAB-D, it meets all the requirements described in this section.

Therefore, the co-simulation framework used in this project consists mainly of GridLAB-D and Jade. As evidenced by the experimental work carried out in Chapter 6 and Chapter 7, this combination has proven adequate and effective in demonstrating the theoretical research conducted throughout this work. A more in-depth study of what the best co-simulation framework might be, if indeed it exists, is beyond the scope of this dissertation.

5.2 Infrastructure overview

In GridLAB-D, the scenarios to be simulated are defined in external files by using a particular syntax. These files are commonly known as *GLM* files. In order to support the concepts introduced in this document, including those related with the Agency Services model and the energy markets, the default syntax of GridLAB-D has been extended with a new plug-in called *AgencyServices*.

Both the system operator and the ASPEMs are implemented as Java web applications running in Jetty [Ecl] servers. Each ASPEM is provided with an agents' container in which the corresponding broker agents are executed. As mentioned before, the broker agents are implemented by using the Jade framework.

On the customers' side, each "`house`" element of the scenario is provided with an element of type "`ASBox`", which stands for *Agency Services Box*. This new type is added by the plug-in *AgencyServices*. The ASBox is conceived as a component of the ESI interface [Hol09]. Local agents run inside ASBoxes and receive OpenADR messages sent by the broker agents. Specifically, broker agents send OpenADR messages of type "`OadrDistributeEvent`", which specifies the level of consumption that households must adopt.

Broker agents communicate with the local agents through the XMPP protocol [XMP], which is one of the communication mechanisms proposed by the OpenADR standard. As a result, the simulation infrastructure also includes a XMPP server. To this end, the *OpenFire* software [Ign14] is used. Another important component of the infrastructure is the repository of demand estimates. This is useful because nodes know the amount of energy they will consume in a specific time period for each of the predefined levels of consumption. For the experiments in this project, estimates were calculated as the mean of ten simulations, sampled every minute. All this information has been stored in a *PostgreSQL* database [Gro96].

Figure 5.1 depicts the main components of the simulation infrastructure, as well as the software components used to implement them. The source code and instructions on how to install the infrastructure and run simulations can be found in [ijl14].

5.3 Modules and applications

The implementation of the simulation infrastructure consists of multiple modules and applications. Three of them can be classified as root applications: two of them are Java applications devoted to representing the roles of system operator and ASPEMs, while the third one is the GridLAB-D simulator enriched with the *AgencyServices* plug-in. These applications interact with each other using standard communication protocols related to both agents' world and energy interoperation. In addition, in order to achieve a clean, modular design, the infrastructure includes complementary software packages that provide specialized functionalities to the root applications, thus enabling them to focus the scope of their business logic.

The following sections describe the objective and relevant aspects of all software modules that compose the simulation infrastructure. Figure 5.2 aims to guide the reader through the interactions and dependencies between them. This is an *UML2 components diagram*, in which each component is tagged with *stereotypes* that help to classify them as *application* or *module*, set the programming language

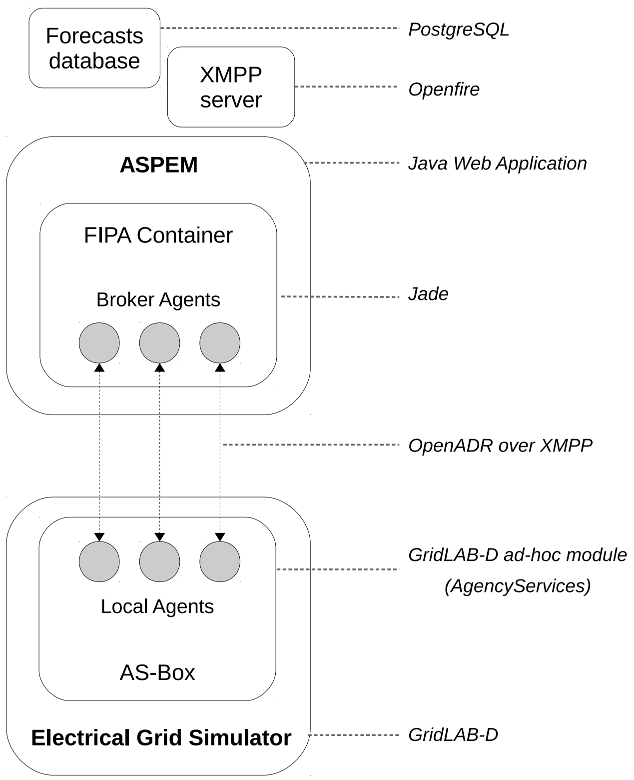

Figure 5.1: Components of the simulation infrastructure.

in which they are developed, and define their authorship. Regarding the latter tag, the value "*Own Development*" means that the component was developed in the framework of this project.

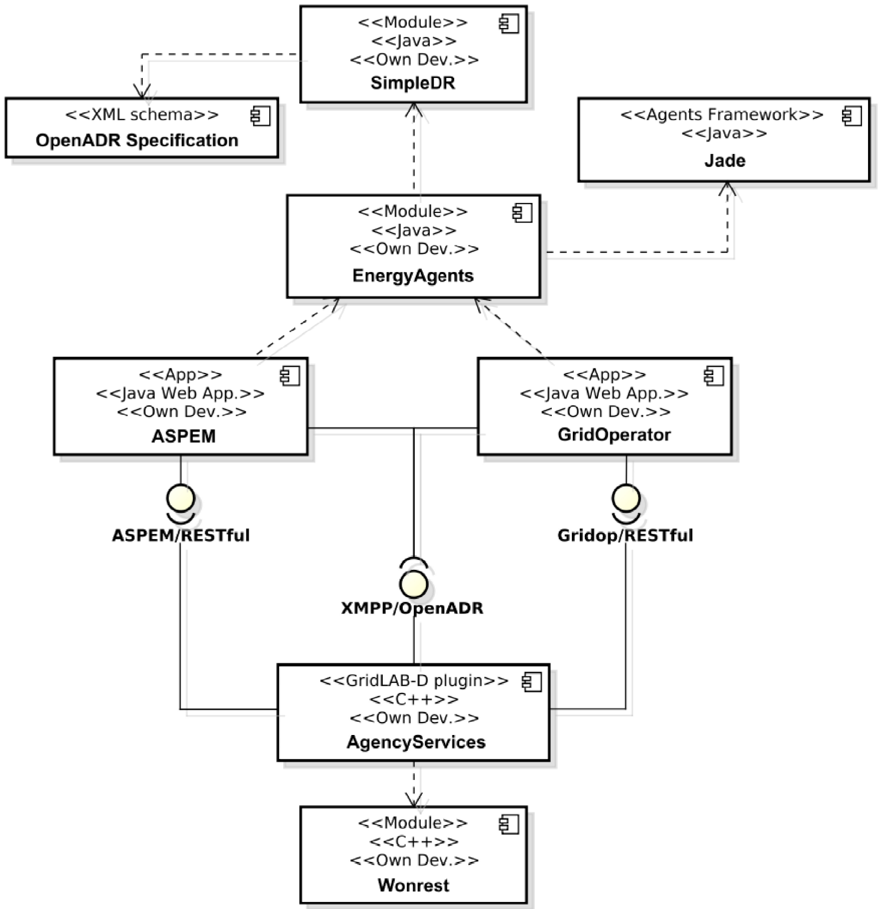

Figure 5.2: UML2 components diagram of the software modules that make up the simulation infrastructure.

5.3.1 **GridLAB-D module**

The plug-in *AgencyServices* extends the vocabulary of GridLAB-D in order to connect it with the layer of software agents. This section is dedicated to the description of the GLM elements that this plug-in adds to the grammar of GridLAB-D.

To use the new functions provided by the plug-in, the GLM file must declare the module, define the system operator, define the ASPEMs, and bind an ASBox element with each household participating in the management system. The following code snippet shows an overview of a GLM file using the new module. Furthermore, the following subsections detail the purpose and parameters of each of the new elements.

```
module agencyservices {
  ...
};
object GridOperator {
  ...
  object ASPEM {
    ...
  };
  object ASPEM {
    ...
  };
};

object house {
  ...
  object ASBox {
    ...
  };
};
```

5.3.1.1 Element `agencyservices`

The statement "`module agencyservices`" is required to load the module into the kernel of GridLAB-D. The following fields must be included in the statement:

- *nodeName*: Code of the test feeder to be simulated. This must be previously registered with the information system of the system operator (see Section 5.3.2).

- *scenarioCode*: Code of the DR program that must be executed over the test feeder specified in the "`nodeName`" parameter. The DR program, which is identified by a unique code, must also be registered with the information system of the system operator. The syntax of the files used to define DR programs is explained in the Section 5.4.1.
- *xmppUrl*: URL of the host running the XMPP server. It is assumed that the server is accessible via the default ports, which are 5222 and 5223.
- *maxStepTime*: Maximum time that the simulation can be running without being interrupted by the module. After this time, the module is invoked to check if all units are synchronized. This parameter is not required, being especially meant for debugging.
- *oadrLevel*: Level of consumption that units must adopt when no DR signal is applied. Supported values are: 0 (*normal*), 1 (*moderate*), 2 (*high*) and 3 (*critical*). This parameter is not required, and the value 0 is taken by default.

The following code snippet shows a basic statement to declare the module:

```
module agencyservices {
  nodename "ieee13";
  scenarioCode "noon-dr-event";
  xmppUrl "localhost";
  maxStepTime 7200;
  oadrLevel 0;
};
```

5.3.1.2 Element `GridOperator`

The "`GridOperator`" element is intended to connect GridLAB-D with the information system of the system operator, which is implemented as a Java web application (see Section 5.3.2) accessible through a web services API. The element "`GridOperator`" also contains the definition of the ASPEM elements participating in the simulation, whose type is described in more detail in Section 5.3.1.3. The fields that "`GridOperator`" must contain are:

- *name*: Internal name of the element. This field is required in order that other elements can refer to the "`GridOperator`" object. For instance, ASPEMs use this token to register with the system operator.
- *servicesEndpoint*: URL to access the web services API.

The following code snippet shows a basic statement of this type of element:

```
object GridOperator {
 name "gridop";
 servicesEndpoint "localhost:9090/gridop/resources";
};
```

5.3.1.3 Element **ASPEM**

The "`ASPEM`" element defines an ASPEM. As in the case of the system operator, it is a Java web application whose functionalities are accessible through a web services API. It must be noted that "`ASPEM`" elements must be declared as inner objects of "`GridOperator`". The fields that the "`ASPEM`" element includes are:

- *name*: Internal name of the element. This name is required in order for other elements can refer to the ASPEM. For instance, households use this token to define the ASPEM to which they are connected via the ASBox.
- *parent*: Internal name of the "`GridOperator`" element.
- *servicesEndpoint*: URL to access the web services API.

The following code snippet shows a basic statement of an "`ASPEM`" element:

```
object ASPEM {
 name "aspem01";
 parent "gridop";
 servicesEndpoint "localhost:8080/aspem/resources";
};
```

5.3.1.4 Element **ASBox**

In order that a household can participate in the management system, it must contain an element of type "`ASBox`". This is responsible for communicating with the corresponding ASPEM, and thus being able to apply the orders sent by the broker agents, as well as sending information on the state of the local resources. This element is also used to set user preferences, including the roles they will play in the markets, and characteristics related to their behaviors. The fields of "`ASBox`" objects are:

- *name*: Internal name of the element. ASBoxes are end nodes (no other element refers to them). However, specifying the internal name is still important for debugging and logging activity.
- *parent*: Internal name of the "`house`" element to which the ASBox is linked.
- *aspem*: Internal name of the "`ASPEM`" element to which the ASBox is connected.
- *levelsSchedule*: Internal name of the schedule element that associates values to time periods. The value corresponding to a period identifies: (*i*) the mode of operation of the household (*hard*, *easy* or *normal*); and (*ii*) the level of consumption when the selected mode is *hard* or *easy*, which can be *normal*, *moderate*, *high* or *critical*. Internally, these two values are combined into a number; however, for the sake of readability, henceforth this combination is shown in plain text. As for schedules definition, GridLAB-D provides a native type that uses the same syntax and semantic as the well-known piece of software *Crontab* [Rez93].
- *startingPrice*: Maximum price [1] at which the broker agent is willing to buy energy blocks in parallel auction markets.
- *levelPriceX*: Price at which the broker agent is willing to sell energy blocks corresponding to the *moderate* (`levelPrice1`), *high* (`levelPrice2`) and *critical* (`levelPrice3`) level of consumption in energy markets.
- *priority*: Integer value useful for resolving conflicts and/or prioritizing some customers over others. The value of this parameter is meant to be contracted with the ASPEM.

The following code snippet shows a basic statement of an "`ASBox`" element:

[1] The market is based on **reverse** auctions, so the definition of concepts such as the starting price is inverted.

```
object ASBox {
  name i1B645;
  aspem aspem01;
  parent house1B_tm_B_1_645;
  levelsSchedule sch_02;
  startingPrice 140;
  levelPrice1 105;
  levelPrice2 140;
  levelPrice3 143;
  priority 3;
};
```

As an example, the code snippet shown below contains a GridLAB-D schedule with three periods for the August 1st. It can be seen that from 13:30h to 15:30h the node works as a hard-load, willing to protect the *moderate* level as minimum; whereas from 15:30h to 17:00h, it is willing to adopt the *high* level even if it is not required. Thereafter, the node adopts the *normal* mode. It is assumed that the node works in *normal* mode for all those periods that are not explicitly defined.

```
schedule sch_02 {
  30 13 1 8 * hard(moderate);
  30 15 1 8 * easy(high);
  00 17 1 8 * normal;
};
```

5.3.2 System Operator application

The system operator is represented by a Java web application running in a Jetty server. The application provides the following functionalities: (*i*) a user interface for easy configuration of scenarios; (*ii*) instantiation of the software agent that carries out the system operator tasks, including life cycle management of DR events; and (*iii*) web services API that GridLAB-D uses to communicate with the system operator. The implementation and instantiation of the software agent is addressed using the module *EnergyAgents* (see Section 5.3.4).

The primary objective of the user interface is to provide an easy means whereby users can configure the scenarios available. Specifically, scenarios are created through the form *New Scenario* (Figure 5.3), which requests the following information:

- *Code*: Unique code related to the scenario being created. When GridLAB-D communicates with the system operator application, this code is used to identify the scenario being simulated (see Section 5.3.1.1).
- *Store name*: Name of the database of demand estimates to be used in simulations.
- *Program*: File with the definition of the DR program to be applied during the simulation (see Section 5.4.1).
- *Description*: User comments about the scenario.

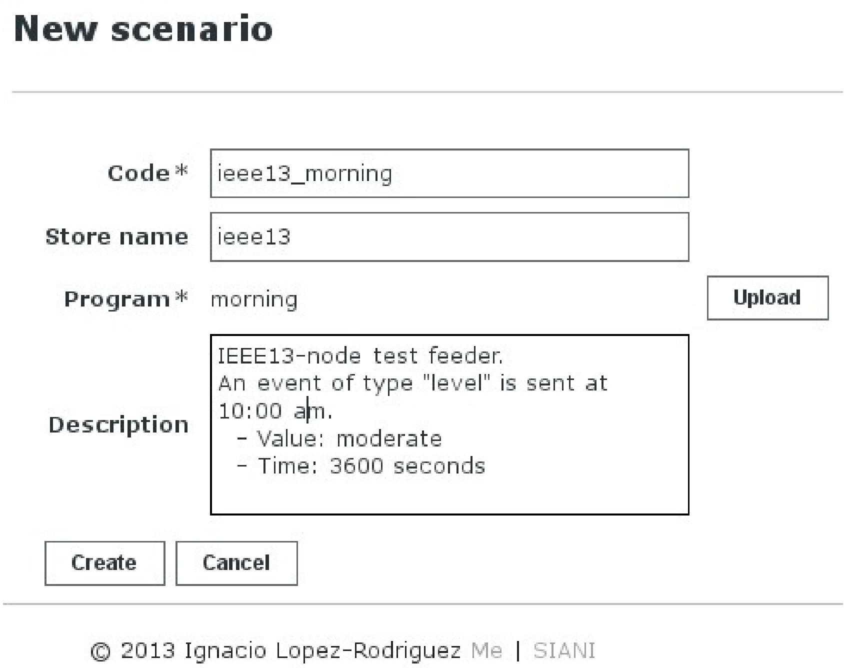

Figure 5.3: Form to register a new scenario.

Moreover, the application provides views for listing the current scenarios and providing information on the simulations performed so far. These views include basic actions that allow users to create, modify and delete items.

Table 5.1 describes the RESTful web services API provided by this application. As can be seen, services are related to the management of the simulation process. Essentially, GridLAB-D uses this set of services to inform the system operator of simulation events, and register new ASPEMs when a new simulation is started.

/services/simulations

Name	**URL**	**Oper.**	**Description**	**Parameters**
start	/	POST	Start a new simulation, and return the identifier related to the simulation.	`scenariocode`: Code of the scenario being simulated. `usp`: Flag indicating whether using starting prices (1) or not (0). `rnd`: Flag indicating whether using uniform distribution of buyers (1) or not (0) (see Chapter 7). `cf`: Value of the constant C_f (see Chapter 7).
finish	{id}/finish	PUT	End the simulation related to the identifier `id`.	`id`: Identifier of the simulation.
pause	{id}/paused	PUT	Pause the simulation related to the identifier `id`.	`id`: Identifier of the simulation.
newAspem	{id}/aspem	POST	Instantiate a new ASPEM, and return the identifier related to the ASPEM.	`id`: Identifier of the simulation. `content`: XML description of the ASPEM (see Listing A.1, Appendix A).

Table 5.1: Description of the layer of RESTful web services provided by the application that represents the system operator.

5.3.3 ASPEM application

ASPEMs are implemented as Java web applications that provide the following functionalities: (*i*) a user interface; (*ii*) web services API that GridLAB-D uses to communicate with the ASPEM; (*iii*) instantiation of the software agent responsible for handling the DR events; (*iv*) creation of the virtual environment in which negotiations are performed; and (*v*) instantiation and deployment of the broker agents that act on behalf of users. The creation and activity of software agents, including the environment they run, is addressed through the module *EnergyAgents* (see Section 5.3.4).

For ASPEMs, the user interface does not provide great functionality, since broker agents as well as other information that might be of interest, are dynamically loaded when the simulation starts. Therefore, the user interface only contains an information screen that allows users to check if the application is properly working.

On the other hand, Table 5.2 describes the RESTful web services API of the ASPEM application. This is meant to inform the ASPEM when simulations are initiated and completed. In addition, once a simulation is running, it offers a service through which GridLAB-D registers the ASBoxes that will be connected to a particular ASPEM. This service receives user preferences as a parameter. For instance, in the particular case of auctions market, for each ASBox, GridLAB-D reports the ASPEM when the user participates as easy-load and hard-load, the price at which he/she is willing to buy or sell energy, and the starting price when he/she acts as auctioneer. Moreover, the RESTful API provides a service that reports the status of the ASPEM. Thanks to this, the system operator can check if ASPEMs are operating as expected.

5.3.4 **Module EnergyAgents**

The *EnergyAgents* module implements all the functionalities related to software agents that both the system operator and ASPEMs require. This module therefore works as an external API that frees root applications from dealing with specialized concepts and procedures belonging to the artificial intelligence area. *EnergyAgents* is, in turn, built on the Jade framework, which facilitates the implementation of multi-agent systems fully compliant with the FIPA specifications [FIP96].

The module creates a platform composed of multiple containers [FIP03a] which work as running environments for the software agents. Following the Agency Services model's guidelines, the system operator and each ASPEM owns a container. Furthermore, for each management method used in simulations, the *EnergyAgents* module implements all behaviors of: (*i*) the agent which acts on behalf of the system operator responsible for managing the event's life cycle; (*ii*) the agent that

Name	URL	Oper.	Description	Parameters
/services/simulation				
start	`/{id}/start`	`POST`	Report that a new simulation has been initiated.	`id`: Identifier of the simulation.
finish	`/{id}/finish`	`PUT`	Report that the simulation related to the identifier `id` is complete.	`id`: Identifier of the simulation.
newASBox	`/{id}/asbox`	`POST`	Instantiate a new ASBox inside the ASPEM and relate it to the simulation with identifier `id`.	`id`: Identifier of the simulation. `clientcode`: Code that identifies the ASBox in GridLAB-D. `content`: XML description of the user preferences (see Listing A.2, Appendix A).
/services/aspem				
status	`/id/status`	`GET`	Return whether the ASPEM is available or not.	

Table 5.2: Description of the layer of RESTful web services provided by the application that represents ASPEMs.

represents a particular ASPEM, which, after receiving the order to handle a particular event from the system operator, decides how to implement it; and (*iii*) the broker agents that participate in the management system when it is implemented as a distributed mechanism.

In order to achieve a well-defined solution, the module defines ontologies that implement all the concepts, actions and predicates involved in the interactions and dialogues between agents. Specifically, an ontology is defined for each management method used in the experimental evaluation. In addition, all interactions between broker agents are conducted through standard FIPA interaction protocols. Specifically, the *FIPA Request* [FIP02c] is used when one agent requests another to perform an action, the *FIPA Query* [FIP02b] for asking for specific information, and the *FIPA Inform* [FIP01] communicate act for sending information about states and facts.

In the source code, both ontologies and agents' behaviors are grouped in Java *packages* so that each of these is associated with a specific method and experiment.

The correspondence between packages and management methods is noted in the code itself.

The *EnergyAgents* module also creates its own layer of entities and functionalities, which, in addition to hiding the interface of Jade, results in a highly simplified interaction layer. Proof of this is the code of the system operator and ASPEMs applications, whose tasks concerning agents consist of a few calls.

5.3.5 Additional modules

The simulation infrastructure provides two additional modules that aim to facilitate specialized tasks. These are: (*i*) *SimpleDR*, which facilitates the creation and management of entities and functions typical of the OpenADR standard; and (*ii*) *Wonrest*, which provides an API for accessing web services of type RESTful using the C++ programming language.

To work with OpenADR it is necessary to extract the entities model from the XML schemes provided by the standard. The generated model is rather complex, so, in practice, using it leads to code difficult to maintain and read. The *SimpleDR* project aims to overcome this problem by providing a much more simplified model of entities. This new model successfully meets all requirements of the experimental evaluation, freeing the developer from having to manage many entities, which are automatically filled. On the other hand, the *Wonrest* library implements a simple RESTful client for the C++ programming language. It does not pretend to be a full implementation of the RESTful specification, but a simple interface for essential functions, hiding much of the complexity that involves working directly with the *Curl* library [cur15], which is the classical approach.

Thanks to these complementary software modules, the business logic of both the *EnergyAgents* and the *AgencyServices* modules is actually focused on the issues they really have to solve.

5.4 Data files

5.4.1 DR programs file

The DR programs are defined through XML files. The definition of *DR program* used in this project consists of zero o more *events*, which in turn may consist of one or more *intervals*. The *event* element also specifies, in form of attribute, the type of signal to be sent, and the starting time. As for *interval* elements are intended to specify the value of the signal for a specific time period.

The following code snippet shows the definition of a DR program which consists of two events. The first event declares a signal of type *delta* that starts at 14:00h. This signal is implemented by using two intervals. The first one lasts 3600 seconds and specifies that 1000 kW must be discarded; while the second interval lasts 1800 seconds and specifies a value of 2000 kW. On the other hand, the second event declares a signal of type *simple* that starts at 15:30h. This consists of three events, each one lasting 1800 seconds. Respectively, they order the customers to adopt the levels *moderate* (1), *high* (2) and *critical* (3).

```
<?xml version="1.0" encoding="UTF-8" ?>
<program xmlns="http://www.siani.es/agencyservices/events/1.0"
   name="h14-d3500">
  <event type="delta" start="2000-08-01T14:00:00Z" priority="0"
     notifDuration="300">
    <interval duration="3600" value="1000" />
    <interval duration="1800" value="2000" />
  </event>

  <event type="level" start="2000-08-01T15:30:00Z" priority="0"
     notifDuration="300">
    <interval duration="1800" value="1" />
    <interval duration="1800" value="2" />
    <interval duration="1800" value="3" />
  </event>
</program>
```

The elements containing a file defining a DR program are:

- *program*: Main element of the XML document. It may contain zero or more elements of type “`event`”. The attributes of “`program`” are:

  - *xmlns*: *Namespace* associated with the XML elements. It must be the same as that defined in the previous example.
  - *name*: Internal name of the program. This name is shown by the Java application of the system operator in order that users can identify the programs.
- *event*: An OpenADR event. An event consists of one or more intervals. Intervals are supposed to span consecutive time blocks, so the duration of the event is equal to the sum of all the intervals duration. The attributes of "`event`" are:
  - *type*: Type of the OpenADR event. The possible values are *level* and *delta* (see Section 2.3.2, page 34).
  - *start*: Time when the OpenADR event starts.
  - *priority*: Integer value that defines the event's importance. In the simulation infrastructure presented in this document, when *priority* is higher than zero, ASPEMs are required to send the signal directly to households, thus indicating that markets cannot be used.
  - *notifDuration*: Time slot within which events must be notified to households.
- *interval*: Element that associates a particular value to a time period of the covering event. The attributes of "`interval`" are:
  - *duration*: Interval duration in seconds.
  - *value*: Value associated to the block of time spanned by the interval. If the type of the covering event is *delta*, the value expresses kW; whereas if the type is *level*, the possible values are: 0 (*normal*), 1 (*moderate*), 2 (*high*) and 3 (*critical*).

This syntax means a significant simplification compared to that provided by the OpenADR standard. For instance, the latter offers the possibility of setting the type of signal in the `interval` element. Although this feature is not supported, the same result can be achieved with the new syntax by defining events for each interval. The main objective of this new syntax is the search for simplicity and clarity without missing important functionalities for the experimental evaluation of the management methods.

5.4.2 Simulation results file

As a result of a simulation, the module *EnergyAgents* generates an XML file that contains information related to the market's activity. The root element is "simulationResults". This is composed of a sequence of elements of type "market" intended to describe the activity of each market involved in the simulation. Note that markets have a predefined duration, so a single event might require conducting multiple markets. The "market" element includes a sequence of broker actions which are defined via the "brokerAction" element. In form of attributes, this element defines the identifier of the broker agent related to the action, and the signal level applied as result of it. On the other hand, the content of "brokerAction" varies according to the role played by the agent in the market, which can be consumer, producer or neither of the two. In particular, the latter behavior is specified by using the "normal" element.

The following code snippet describes a simulation containing a market in which two broker agents do not participate either as consumers or as producers.

```
<?xml version="1.0" encoding="UTF-8" ?>
<simulationResults xmlns="http://www.siani.es/agencyservices/
    simulationResults/1.0" scenarioCode="ieee13" nMarkets="1"
    usingStartingPrice="true" usingRandomDistribution="true" cteCF
    ="1">
  <market start="2000-08-01T15:30:00Z">
    <brokerAction id="i48A671" level="1">
      <normal />
    </brokerAction>
    <brokerAction id="i2A652" level="1">
      <normal />
    </brokerAction>
  </market>
</simulationResults>
```

When the broker plays the role of producer, the "producer" element is used instead. In the attributes section, it specifies the generation capacity and the amount of this that was successfully placed in the market. In addition, this information is further detailed with a sequence of inner elements of type "genBlock". Each of them describes the price at which a particular block of energy was placed, and to whom it was sold. The following code snippet shows an example of a producer's action:

```
<brokerAction id="i48A671" level="3">
  <producer capacity="1000" placed="1000">
    <genBlock amount="500" price="100" to="i67C671" />
    <genBlock amount="500" price="300" to="i25A652" />
  </producer>
</brokerAction>
```

On the other hand, the element "consumer" is used when the broker works as a consumer in the market. In the attributes section, it defines the amount demanded by the broker agent, and the part of this that was successfully covered in the market. This information is further detailed through inner elements of type "loadBlock". Each of these inner elements specifies the price at which a specific block of energy was bought, and from whom it was bought.

```
<brokerAction id="i48A671" level="3">
  <consumer demanded="1000" covered="1000">
    <loadBlock amount="500" price="100" from="i67C671" />
    <loadBlock amount="500" price="300" from="i25A652" />
  </consumer>
</brokerAction>
```

As shown in the first code snippet, the root element "simulationResults" contains attributes that are useful to contextualize the experiment. Inter alia, they include the code of the simulated scenario and parameters related to the algorithm used to implement the market. A full description of the syntax of the XML can be found in Listing A.3, Appendix A.

Another important source of information for evaluating the results of the experiments is the output data generated by GridLAB-D. Its syntax provides the element "collector", which is able to store the state of objects in text files. For the experiments developed in the context of this project, *collectors* were added to obtain detailed information about households, water-heaters, HVACS and lights. For instance, the following code snippet describes a typical statement to collect load data from all objects of type "house".

```
object collector {
   file "${target_path}/houses-load.csv";
   group "class=house";
   property sum(total_load),avg(total_load);
   interval 0;
   limit 0;
};
```

5.5 Simulation life cycle

The need for using a co-simulation infrastructure, thus involving several modules that interact with each other, means the simulation life cycle is not an obvious process. This section aims to shed light on it. The explanation, for the sake of clarity, is divided into three main stages: *initiation*, *execution* and *completion*.

In the **initiation stage** (Figure 5.4) all components of the Agency Services model located at the cloud are initialized. Specifically, they are the nodes corresponding to the system operator, ASPEMs and broker agents. The actions performed in this stage are:

1.1. GridLAB-D informs the system operator's application (henceforth *GridopApp*) that a new simulation has been started. This action is performed using the web service `start` of *GridopApp* (see Table 5.1, page 108).

1.2. GridLAB-D registers with the *GridopApp* all ASPEMs defined in the GLM file. This action is performed using the service "`newAspem`". As a result of this call, *GridopApp* instantiates a new instance of the ASPEM's application (henceforth *AspemApp*), which at implementation level means the initiation of a new instance of the Java server running *AspemApp*.

1.3. GridLAB-D informs each *AspemApp* that a new simulation has been started. The service "`start`" of *AspemApp* is used to perform this action (see Table 5.2, page 110).

1.4. Once all *AspemApps* are operational, GridLAB-D registers each ASBox with its corresponding ASPEM. This action is carried out using the service "`new`

ASBox" of *AspemApp*. As described in Section 5.3.1.3, the service's call contains the users' preferences, including the information on how they will behave in markets.

In the **execution stage** (Figure 5.5), GridLAB-D informs the *GridopApp* when a new event is triggered, and when it is paused to wait for the reply of the ASPEMs and the broker agents. Here it is worth mentioning that GridLAB-D is an asynchronous (event-driven) simulator so that the simulation time advances as it is realized in the implemented model [Mar98].

2.1. GridLAB-D informs *GridopApp* that the simulation is paused. The application, if found necessary, processes the event and indicates the next time at which it should be called again. Specifically, the simulation event is processed if the pause time coincides with the starting time of the next event of the DR program being simulated. GridLAB-D communicates the pause using the web service "pause" of *GridopApp*. When *GridopApp* receives a pause event, it checks that both simulators Jade and GridLAB-D are synchronized (meaning there are no inconsistencies between the times held by both simulators), which is a critical issue when working with co-simulation infrastructures. Note that the following actions are only performed under the assumption that a DR event must be processed.

2.2. *GridopApp* informs all ASPEMs that a new DR event has been launched. The interaction is performed using the software agents that represent both types of entities, so web services are not used, but rather the communication system of the Jade framework.

2.3. Internally, *AspemApp* handles the event according to the method being simulated. Chapter 6 describes two different approaches, although others are possible.

2.4. Once *AspemApp* reaches a solution, each broker agent communicates to its corresponding local agent the action it should apply to the local resources. This interaction is done via the XMPP protocol, which in this case transports messages typical of the OpenADR standard.

2.5. *AspemApp* informs *GridopApp* that the event has been processed. The latter responds to GridLAB-D indicating the next time that, at minimum, it should be invoked again. This value corresponds with the starting time of the next event of the DR program, or "*undefined*" if no more events are programmed.

In the **completion stage** (Figure 5.6), GridLAB-D informs both the ASPEMs and *GridopApp* that the simulation is over in order that they can properly close the involved resources.

3.1. GridLAB-D informs *GridopApp* that the simulation is over using the service "`finished`".

3.2. GridLAB-D informs each *AspemApp* that the simulation is over using the service "`finished`".

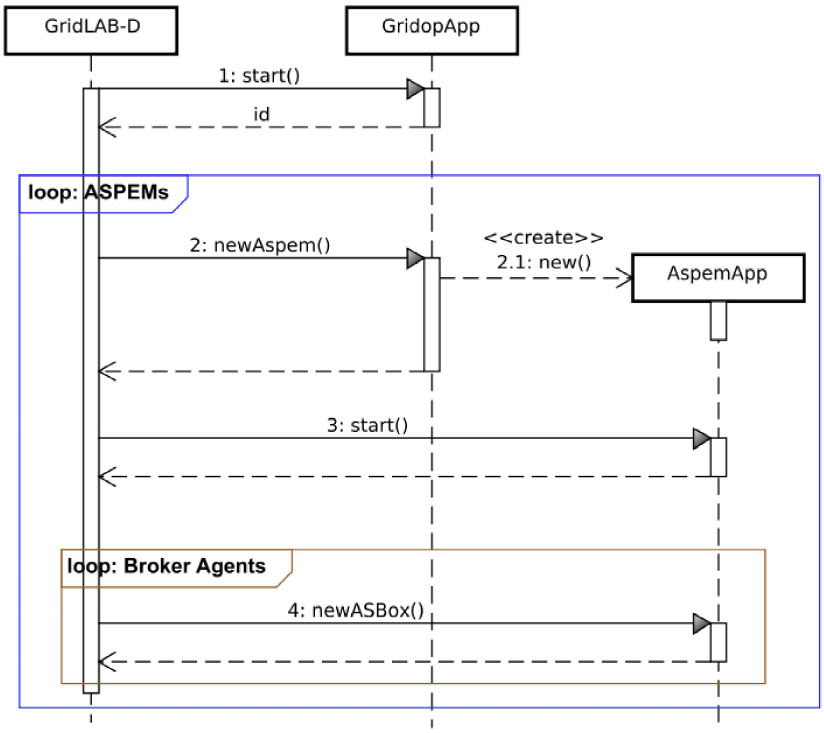

Figure 5.4: UML2 sequence diagram of the initiation stage of the simulation life cycle.

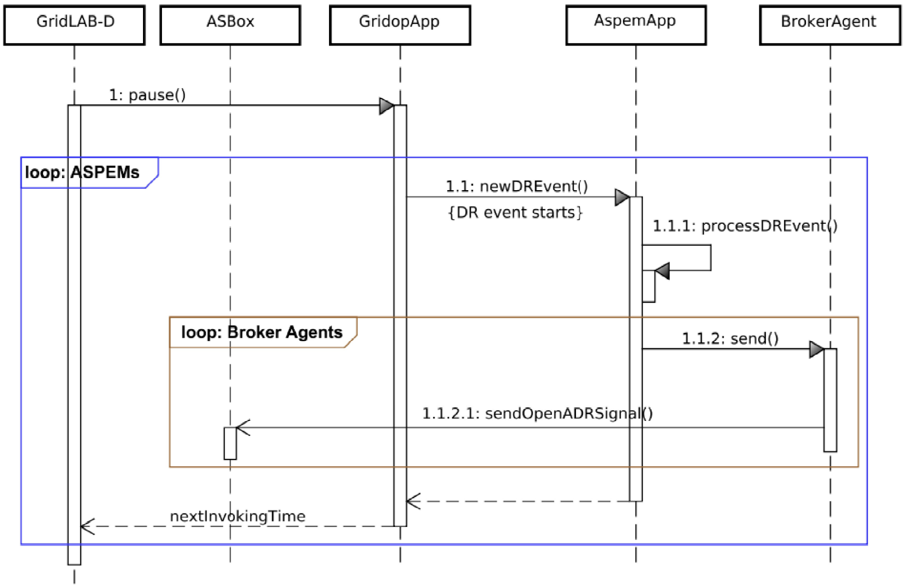

Figure 5.5: UML2 sequence diagram of the execution stage of the simulation life cycle.

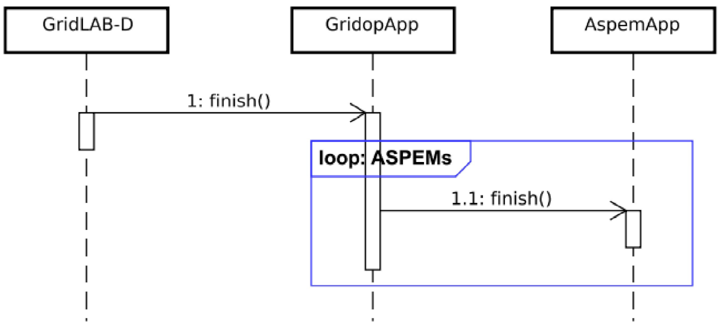

Figure 5.6: UML2 sequence diagram of the completion stage of the simulation life cycle.

– Y la naturaleza ¿también es una tontería? — pronunció Arkadi mirando pensativo a lo lejos, a los campos abigarrados, que el sol ya en declive iluminaba hermosa y suavemente[. . .]
— También lo es, en el sentido que tú le das. La naturaleza no es un templo, sino un taller, y el hombre es un trabajador del taller.

"*Padres e hijos*", Iván Turguénev.

CHAPTER

6

Experimental evaluation of the ASPEM role

The ASPEM role introduced in Chapter 3, as discussed, is designed to work as a virtual environment in which software agents can conduct negotiations on behalf of customers. This concept represents a significant improvement over the classical architecture, in which end-nodes are supposed to be able to perform all type of functions. Specifically, in the context of the Smart Grid, besides managing local resources, they are also expected to access external data services, process data sets and conduct dialogues and negotiations. To some extent, this is the reason why real solutions do not incorporate advanced functionalities, but others which are more practical and less ambitious. In contrast, the ASPEM role, as a characteristic inherited from the Cloud Computing model, promises to free end-nodes from demanding tasks. When it comes to OpenADR, this feature implies that end-nodes should only have to support the simplest profile of the protocol (profile *2.0a*) while still enjoying the functionalities and benefits of the most advanced profiles (*2.0b* and *2.0c*). One of the goals of the present Chapter, which shows the results of simulating the participation of ASPEM nodes in OpenADR programs, is to demonstrate this **conversion skill**: to conduct the management of the grid by sending signals of the simplest profile, regardless of the profile of the signal sent by the system operator.

With the aim of demonstrating the benefits that using software agents can bring to the customers, the simulations are implemented with **parallel auction markets** [PJ05]. As discussed in Chapter 4, this approach meets all the conditions of energy markets (see Section 4.1, page 74); and, not least of all, it is completely distributed, so it is closer to the future vision of the electrical grid. In particular, parallel auctions give the users autonomy and control over the decision process. However, this approach is known to shift the burden of holding auctions from the system operator to the customers. This Chapter also fulfills the mission of showing how this challenge can be met by using the concept of broker agent, which is a software agent contracted as a Cloud Computing service. In this regard, it is worth mentioning that this research work is the first of its kind to simulate parallel auction markets in the context of the electrical grid, thus giving an insight into the actual effectiveness of the mechanism.

The simulations are performed in the context of OpenADR programs, this technology being one of the most immediate and realistic milestones of the Smart Grid. To achieve markets in **DR programs**, which by default is a context in which end-nodes are expected to merely apply incoming signals, the model based on *easy-loads* and *hard-loads* described in Chapter 2 is used (see Section 3.5, page 69). Furthermore, choosing DR programs as simulation context demonstrates the ability of ASPEM nodes to display their capabilities in restrictive environments.

6.1 Experimental evaluation

Two types of experiments are carried out below: the first one consists of a centralized approach that prioritizes customers' preferences according to their contract with the ASPEM, thus focusing on the services facet of the ASPEM nodes; while the second puts forward a decentralized solution that shows the capability of ASPEM nodes to manage the events through energy markets. In both cases, the process consists of the following general steps:

i. The system operator sends curtailment events. These can be defined as signals of type *simple* or *delta*. In the case of *simple* signals, the system operator expects that nodes adopt one of the following predefined OpenADR levels: *nor-*

mal, *moderate*, *high* or *critical*. As for *delta* signals, they express an amount of load that must be discarded by each ASPEM.

ii. ASPEM processes curtailment events. In the solution based on the users' priority, the ASPEMs select the nodes that participate in the event, as well as the level of consumption they must apply to their devices. On the other hand, when markets are used, hard-loads initiate negotiations with easy-loads in order to avoid applying the level of consumption specified in the signal.

iii. Customers receive signals of type simple from the ASPEM. The local agent applies the actions corresponding to the level indicated by the signal sent by the corresponding ASPEM. These actions consist of switching off devices.

In the experiments, OpenADR levels are implemented as follows:

- *Normal*: No restriction is applied, so the household can consume as usual.
- *Moderate*: HVAC units must be switched off.
- *High*: HVAC units and water-heaters must be switched off.
- *Critical*: All units, including lights, must be switched off.

This definition of levels is too aggressive. Actually, for real scenarios the adoption of solutions that vary the comfort level is commonly proposed. To achieve this, ranges of operating points are set for each of the consumption levels, so users do not lose the entire service provided by the units, but only part of the comfort they provide. However, we prefer the above definition for simulations because it provides demand curves where it is easier to differentiate the application of each DR signal, besides helping to keep the focus on the exchange mechanism.

In both types of experiments, a typical summer day (August 1st, 2000) is simulated in the scenario *IEEE 13-node* [Chr99], which is configured with two ASPEMs that have practically the same number of clients and the same level of consumption per OpenADR level (Table 6.1) and per type of load (Table 6.2). In order to configure the nodes' preferences, three types of schedules are defined (Listing 6.1). They configure the nodes to act as easy-, hard- or normal-loads during the whole period of simulation. In particular, the hard-loads are configured to protect all their demand, so they strive to preserve the normal state of operation; and the easy-loads are willing to discard all the demand. Table 6.3 summarizes the number of nodes associated to each type of schedule.

```
schedule easy {
   0 14 1 8 * easy(critical);
   0 15 1 8 * easy(critical);
   0 16 1 8 * normal;
};

schedule hard {
   0 14 1 8 * hard(normal);
   0 15 1 8 * hard(normal);
   0 16 1 8 * normal;
};

schedule normal {
   0 0 1 8 * normal;
};
```

Listing 6.1: Schedules definition for the auctions market.

	normal (kW)	moderate (kW)	high (kW)	critical (kW)
aspem 1	61.137	27.103	15.552	0
aspem 2	66.461	30.871	15.194	0

Table 6.1: Consumption of the ASPEMs per OpenADR level.

	Easy load (kW)	Hard load (kW)
aspem 1	4.670	32.652
aspem 2	5.819	34.891

Table 6.2: Consumption of the ASPEMs per type of load.

	Normal	Easy	Hard
aspem 1	109	31	174
aspem 2	107	36	172

Table 6.3: Number of nodes associated to each type of schedule.

In absence of any DR event, the behavior of the demand curve is that depicted in Figure 6.1. It shows that there is a peak demand in the afternoon, which is especially high between 14:00h and 15:30h. The following experiments are focused in this particular zone of the curve, to which OpenADR events of type *level* and *delta* are ordered.

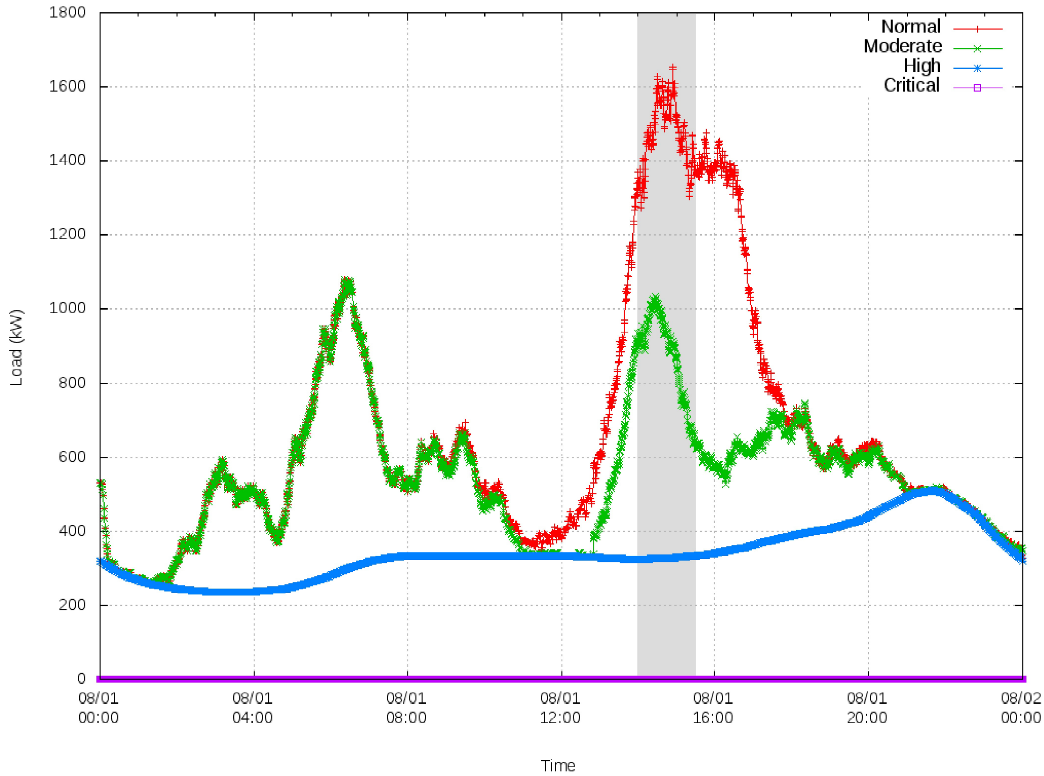

Figure 6.1: Demand curve corresponding to 1st August, 2000 in the IEEE 13-node when no signal is applied.

As will be appreciated, the joint recovery of all the consumption devices after a DR event produces noticeable peaks in the demand curve. This is a well-known effect for which the OpenADR standard provides an operating procedure. However, being beyond the scope of this document, no mechanism has been implemented in the experiments.

It is also worth mentioning that, in order to give the system operator a channel to bypass the action of ASPEMs, all OpenADR messages with priority higher than zero are conveyed intact to the child nodes. This channel can be useful for handling situations that require particular behaviors, such as emergencies.

6.1.1 Dispatch based on priorities

In this experiment, the ASPEM uses an internal centralized algorithm to determine which nodes take on the curtailment job when an OpenADR signal of type *delta* is received. The decision is based on the content of the contract that links the user to the ASPEM. For this specific experiment, we assume that users can specify in their contract:

i. The time periods at which they want to participate as hard-, easy- and normal-loads.

ii. A priority level that the ASPEM uses to determine which nodes are used to meet the curtailment signal. The higher the value of the parameter, the more the preferences defined by the customer are considered.

According to the syntax of the ad-hoc plugin developed for GridLAB-D (see Section 5.3.1, page 101), the above information is expressed by using elements of type "`ASBox`" as follows:

```
object ASBox {
 name i1B645;
 aspem aspem01;
 parent house1B_tm_B_1_645;
 levelsSchedule sch_02;
 priority 3;
};
```

When an OpenADR signal is sent, the ASPEM gathers the previous information by asking the broker agents [1]. Next, the algorithm converts the incoming signal (which in this case is of type *delta*) into multiple signals of type *simple* that are conveyed to the AS-Boxes. It must be noted that OpenADR events are composed of intervals, and therefore the exchange process is carried out for each of these intervals.

The first step of the algorithm is to determine which resources are necessary to cover the amount of load specified in the *delta* signal. This information is obtained by asking each broker agent its demand estimates for each consumption level. Next, depending on this result, three settings are possible:

i. The amount can be entirely covered by using easy-loads (Algorithm 6, Appendix B). This is the less dramatic case for the system, since all the demand to discard is collected from nodes that are actually willing to shed it. In this case, the algorithm sorts the easy-loads in descending order according to the priority value. Next, the algorithm, following the established order, takes the minimum number of loads necessary to cover the amount specified by the *delta* signal.

ii. The amount can be covered without using hard-loads (Algorithm 7, Appendix B). In this case, it is possible to cover the delta amount by doing that no-hard loads adopt one of the predefined levels of the OpenADR standard (*moderate*, *high* or *critical*). The first step in this case is to calculate the level that the no-hard loads have to adopt. This list of nodes is sorted in ascending order according to the priority value. In addition, as in the previous case, a list of easy-loads is built. In order to cover the delta amount, firstly, the list of easy-loads is used; next, the no-hard loads are taken (following the order of the list) until the delta amount is covered.

iii. The amount has to be covered using hard loads (Algorithm 8, Appendix B). Firstly, all the no-hard loads are required to adopt the *critical* level. Next, the list of hard-loads is sorted in ascending order according to the priority value. Nodes from this list are ordered to adopt the critical level until covering the delta amount.

[1] Since this is a centralized solution, the information could also be obtained directly from the database system.

Figure 6.2 illustrates the case in which the operator orders a reduction of 30.000 kW between 14:00h and 15:30h. In this scenario, the operator aims to reduce the required amount and avoid taking the *moderate* level. In general, when the difference of demand between the predefined OpenADR levels is significant, *delta* signals enable to apply reductions with results that are more accurate and require less involvement of clients. Table 6.4 shows that, for handling the event, only easy-loads are used, so there is no need for the nodes working as hard-loads to contribute in the load reduction. Table 6.5 shows the number of signals of each type used to generate the expected result.

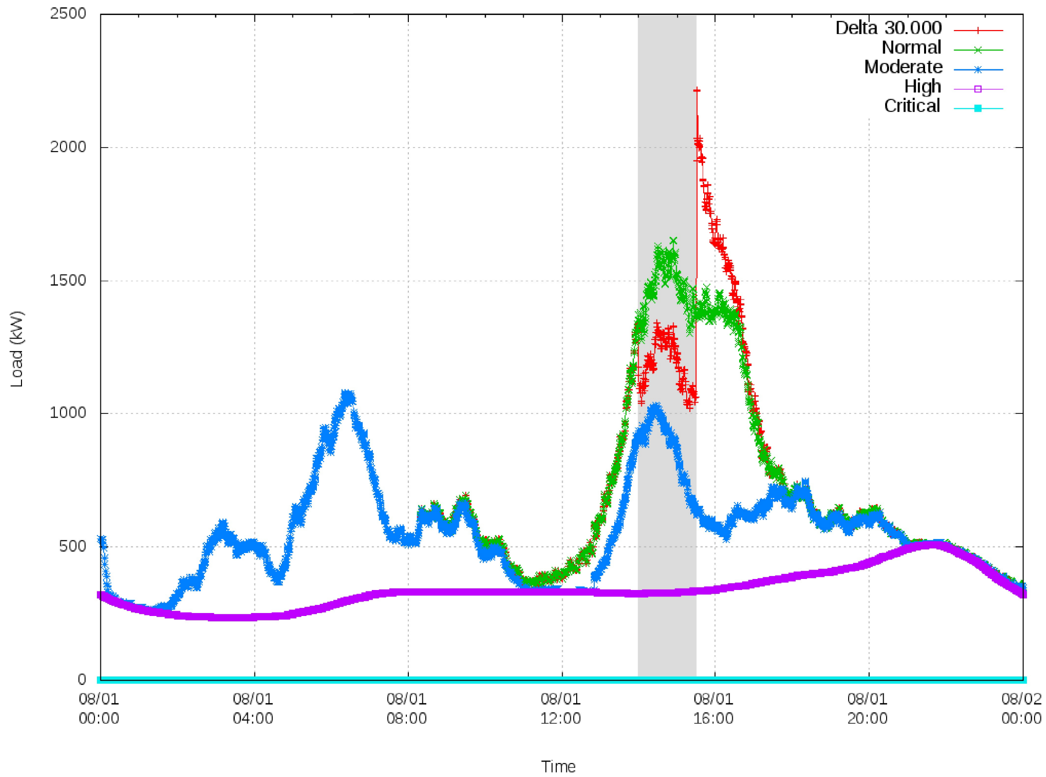

Figure 6.2: Demand curve corresponding to 1st August, 2000 in IEEE 13-node when a delta signal of 30.000 kW is applied.

Moreover, Figure 6.3 illustrates the example in which a *delta* signal of 75.000 kW is applied. It aims to avoid the need for the adoption of the *high* level. Due to

	Delta	**Covered (kW)**	**Normal (kW)**	**Easy (kW)**	**Hard (kW)**
aspem 1	14.465	14.569	9.899	4.670	0
aspem 2	15.534	15.680	9.861	5.819	0

Table 6.4: Consumption of the ASPEMs per level when a delta signal of 30.000 kW is applied.

	Normal	**Moderate**	**High**	**Critical**
aspem 1	187	91	36	0
aspem 2	204	80	31	0

Table 6.5: Number of signals per type when a delta signal of 30.000 kW is applied.

the huge size of the reduction, as shown in Table 6.6, the hard-loads are involved in the action. However, in relation to the total capacity of this type of loads, their participation is actually limited: only the 11$\%$ of them are used in this case, and always after all the capacity corresponding to the normal- and easy-loads has been used. Table 6.7 shows the number of signals per type that are applied in this case. All nodes to which a signal of level *normal* is sent are actually hard-loads that have contracted the maximum type of protection to the corresponding ASPEM.

	Delta	**Covered (kW)**	**Normal (kW)**	**Easy (kW)**	**Hard (kW)**
aspem 1	36.163	36.295	27.996	4.670	3.628
aspem 2	38.836	38.977	29.487	5.819	3.671

Table 6.6: Amount of load per type when a delta signal of 75.000 kW is applied.

In both cases, the charts show that the ASPEM role is able to handle *delta* signals by using user properties, which in this case, for the sake of simplicity, are

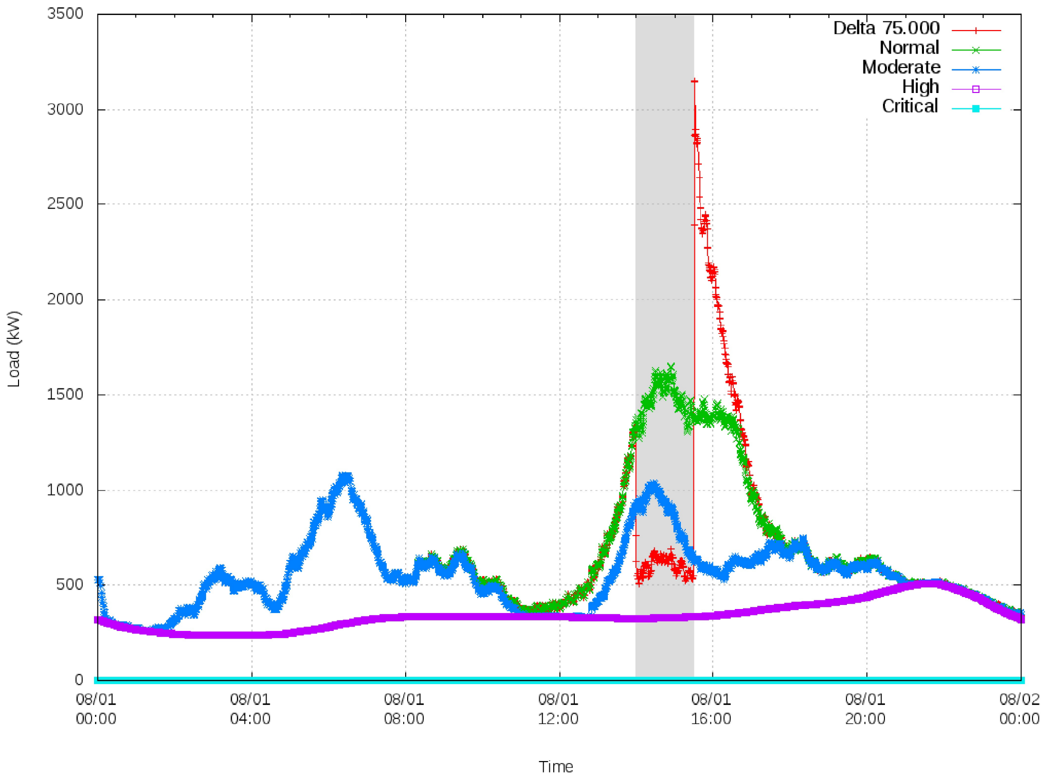

Figure 6.3: Demand curve corresponding to 1st August, 2000 in IEEE 13-node when a delta signal of 70.000 kW is applied.

	Normal	Moderate	High	Critical
aspem 1	128	0	0	186
aspem 2	139	0	0	176

Table 6.7: Number of signals per type when a reduction of 75.000 kW is applied.

crystallized in the form of priorities, so that hard-loads are requested only in extreme situations.

6.1.2 Dispatch based on parallel auction markets

One of the most promising mechanisms for implementing energy exchanges in the Smart Grid are reverse parallel auctions [PJ05, PL06]. In this approach, each node is assumed to have the capacity to hold and participate in auctions in which sellers offer blocks of demand, and producers make bids. These are said to be *reverse* auctions because the seller demands the exchanged good; and are classified as parallel because many of them can be running at the same time, so a producer can bid simultaneously in multiple auctions.

In the auctions market, a user who wants to protect part of her load is a user who offers blocks of demand to be covered (an auctioneer); while a user who is willing to offer part of her load is a user who is willing to bid for covering blocks of demand (a bidder). Therefore, hard-loads are typical of auctioneers, and easy-loads correspond to bidders. Note that the role of an agent may change from market to market, which duration is set to blocks of time of 30 minutes in the following simulations. Therefore, events lasting more than 30 minutes consist of several consecutive markets.

Once an ASPEM receives an OpenADR event from the system operator, the implementation of the auctions market conducted in this section goes through the following stages:

- *Announcement stage*: ASPEMs inform the broker agents about the initiation of new auction markets. Specifically, the message informs the duration of the event, the duration of the markets, and the deadlines of subsequent stages of the process.

- *Registry stage*: All broker agents register with the yellow-pages directory, also known as Directory Facilitator (DF) in multi-agent systems [FIP03a], the role they will play in each market of the event, which can be consumer (auctioneer) or producer (bidder). Each agent registers all the information related to the corresponding role: auctioneers add the starting price; and producers announce the maximum amount of load they can supply.
- *Offering stage*: Auctioneers query the DF in order to find producers and invite them to send bids. Alternatively, producers may also use the DF to find auctions in which they can participate. At the end of this phase, all those auctioneers who do not find enough producers, and all those producers that do not find auctioneers, report this condition to the ASPEM. Their participation is canceled, so they are ordered to apply the original DR signal. It must be noted that all auctioneers that do not find enough producers to cover, as minimum, a full OpenADR level of demand, also have to cancel their auctions. For instance, if a broker agent holds an auction in order to avoid the adoption of the *moderate* level, and the broker agent does not find enough offers to cover the gap that goes from the *normal* level to the *moderate* one, then the auction has to be canceled even before it starts. However, a producer is free to provide part of the level she is giving.
- *Bidding stage*: From all the invitations, the producers decide in which auctions to participate. The offers are sent in form of linear piece-wise functions (Figure 6.4). Each level of the function corresponds to a level of consumption that the producer gives. The number of sections of the function depends on: (*i*) the level of the signal that the system operator sends, which represents the starting point; and (*ii*) the maximum level of the load that the customer is willing to give, which is the maximum level of easy-load. It is important to note that producers normally cannot participate in all auctions because, as the auctions run in parallel, if many of his offers were accepted, the producer would face the risk of overbooking his real capacity. In general, overbooking in parallel auctions is a difficult problem to solve.

- *Clearing stage*: The auctioneers decide which bids are accepted. In this implementation, the clearing algorithm chooses the producers' offers in accordance with the price. To this end, firstly, the algorithm sorts all the bids by price, where a bid is a single section of the piece-wise linear function sent by the producer. Next, bids are accepted iteratively until the demand is covered. As a last step, the auctioneer informs each agent about the refusal or acceptance of his bids. In a real scenario, auctioneers may base their decision on factors other than price, such as reliability, efficiency and pollution level.
- *Committing stage*: The ASPEM receives from brokers the contracts that have been closed and, according to them, sends each broker agent the signal level it must apply, which is subsequently sent to the local agents, and so applied on the local resources. The producers who have accepted offers have to adopt more restrictive levels, while the auctioneers who have closed agreements adopt less restrictive ones. All nodes that do not participate in any exchange, including those that have had to cancel their auctions, will apply the level ordered by the original signal.

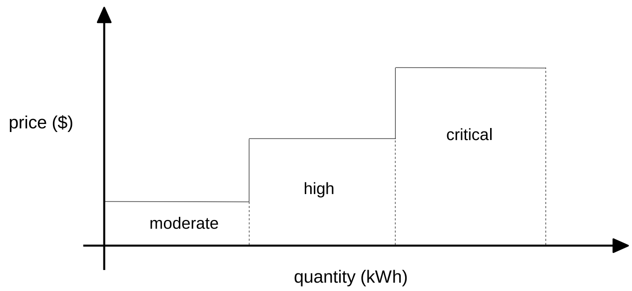

Figure 6.4: Linear piece-wise function used by the bidders in auction markets.

In this section, events are based on signals of type *simple*. In particular, simulations for the cases in which the system operator orders adopting the *moderate* and *high* levels are presented. In both cases, the duration of the event is 90 minutes, starting at 14:00 hrs. and ending at 15:30 hrs., which is a slot of the day with high demand. Moreover, the duration of the markets is set at 30 minutes, so three consecutive auction markets are held during the event.

The schedules defined in the beginning of this section are used in the simulations that follow (Listing 6.1). They were distributed among the households so that the relation of producers (easy schedules) to consumers (hard schedules) is 2.75. That is, the number of producers nearly triples the number of consumers. On the other hand, the price of energy is in the range of $\$80$ to $\$180$. Each agent is assigned a random price function similar to that depicted in Figure 6.4. The starting price is set at random to be slightly higher than the *moderate* level's price, so it cannot be considered a low starting price.

The global profile of the simulated scenario is shown in Table 6.8. According to this, when the value of the signal is *moderate*, the ratio of bidding capacity to amount auctioned (henceforth R_{ca}) is 2.29. Therefore, when auctioneers do not establish starting prices, the offer doubles the demand. The table also shows the ratio of number of producers to number of consumers (henceforth R_{np}), which is 3.10 when the signal is *moderate*. The scenario for the *high* value is less than ideal, since the offer is not enough to cover the demand, being R_{ca} equal to 0.91. Furthermore, it must be noted that both values are smaller when starting prices are present, since this condition implies that an offer may be not valid for part of the auctions.

Signal level	**Auctioned (kW)**	**Bidding capacity (kW)**	R_{ca}	R_{np}
moderate	15271	35114	2.29	3.10
high	20416	18569	0.91	2.27

Table 6.8: Description of the simulated scenarios using parallel auction markets.

As a result of the exchanges between the broker agents, it is expected that the output will be similar to the demand curve corresponding to the level of consumption defined in the DR event. Figure 6.5 illustrates the simulation when a signal with value *moderate* is sent. Figure 6.6 illustrates the curve for a signal with value *high*.

The charts show that the market approach is capable of generating demand curves very similar to the original curves. Furthermore, the definition given of the

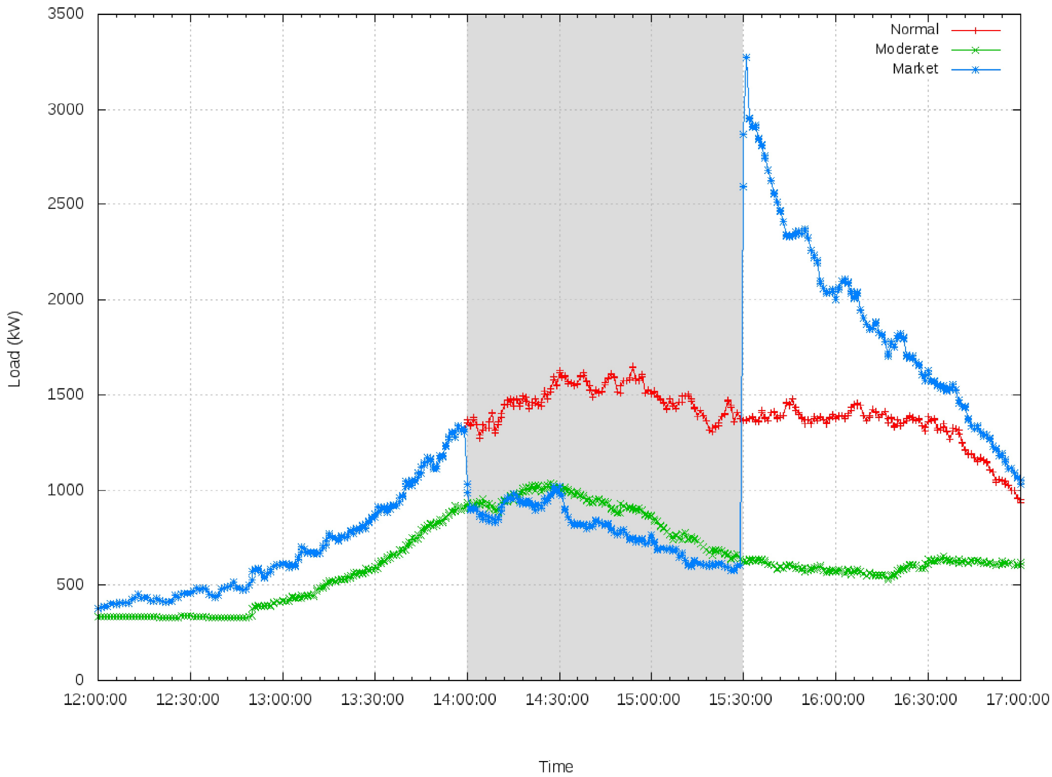

Figure 6.5: Affected part of the demand curve of the parallel auction market (without starting prices) when a moderate signal is ordered.

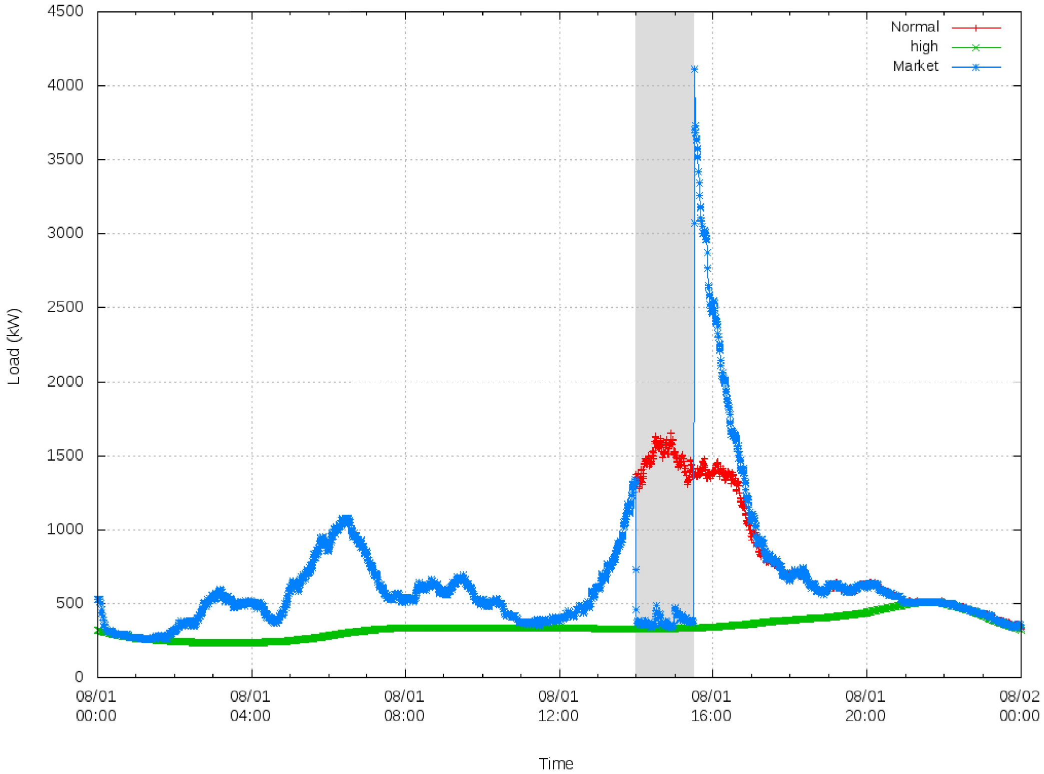

Figure 6.6: Demand curve of the parallel auction market (without starting prices) when a high signal is ordered.

market is safe because when a consumer does not collect enough offers to cover his gap, he is required to cancel the corresponding auction and apply the original signal. This behavior is intended to ensure that the consumption, as a maximum, is equal to the level expected by the system operator. This condition is only threatened by the stabilization conditions of the devices. In this regard, the chart corresponding to the *high* signal shows that, in spite of the response being very similar to the expected one, there are demand peaks that place the output curve slightly above the average values. This is because the *high* level of consumption is implemented by manipulating the activity of water-heaters, whose stabilization period is noticeable for blocks of 30 minutes. However, as commented, this effect can be countered with a better market design. In Figure 6.7, the same scenario is configured so that the duration of the OpenADR event is set to 2 hours and the market duration to 60 minutes. The graph shows that, in doubling the duration of the market, the response curve is practically equal to the expected one. The only demand peak corresponds the end of the first market (after the first 60 minutes) because, as mentioned, no solution for the recovering event has been implemented.

Table 6.9 shows statistics about the simulated markets. The column *Auctions Covered* refers to the percentage of auctions that achieve to cover all their demand and are closed successfully, thus avoiding to apply the signal originally sent by the system operator. As shown, it is noticeable that this percentage is significantly higher when starting prices are disabled; this is because, in this case, each producer's offer is valid for all available auctioneers. However, the most striking fact is that, although many auctions have to be canceled due to the lack of offers, there is a considerable amount of supply that is not used. The reason for this condition is the poor distribution of buyers' participation. Specifically, if many producers choose to participate in the same set of auctions, many auctions will not receive offers (or enough offers) to cover their demand, so they have to be canceled prematurely. In our simulations, producers select randomly the auctions they participate in. This causes that most of the auctions receive offers, but the ratios R_{np} and R_{ca} are not high enough to cause that auctions to be populated with the minimum required number of producers to cover their entire demand. The inefficiency that causes the poor distribution of participants in markets based on parallel auctions have been barely reported in the literature [Hop08].

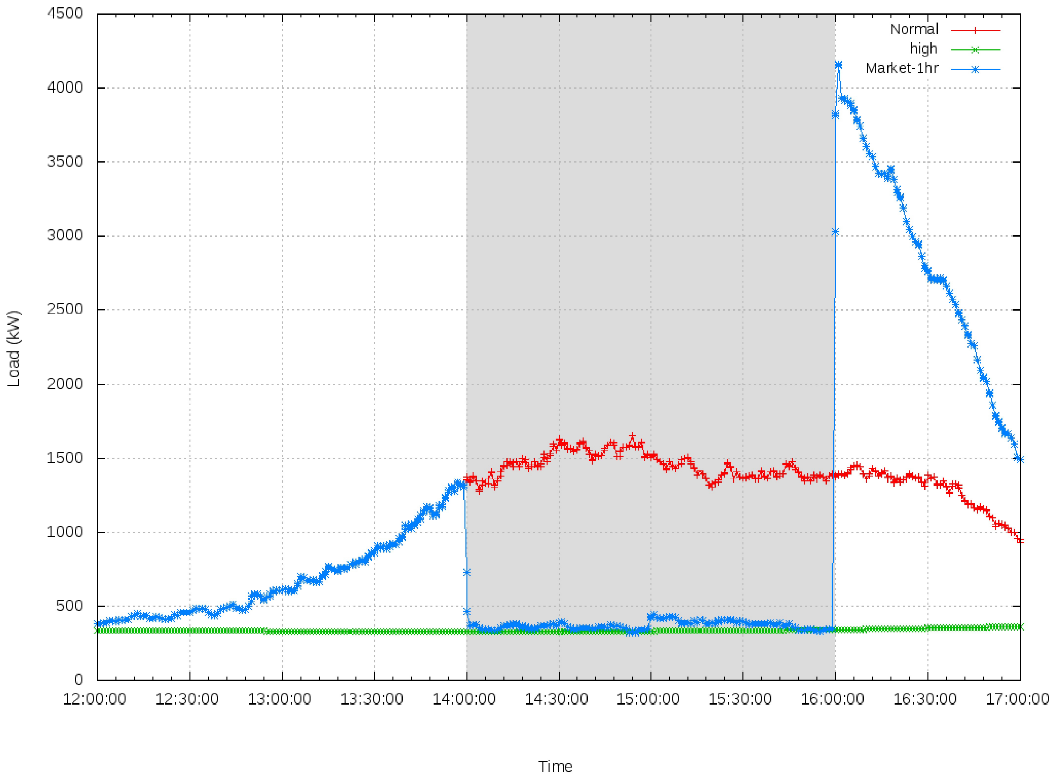

Figure 6.7: Affected part of the demand curve of the parallel auction market when the duration of the market is set to 60 minutes.

	With starting prices			Without starting prices		
	Amount Exchanged (kW)	Auctions Covered (%)	Offer Provided (%)	Amount Exchanged (kW)	Auctions Covered (%)	Offer Provided (%)
moderate	5798	37.97	18.06	12930	84.67	36.82
high	3334	16.33	23.80	3843	18.82	20.70

Table 6.9: Data corresponding to the simulation of parallel auction markets when signals of type *moderate* and *high* are applied.

6.2 Discussion

The ASPEM role offers the possibility of successfully managing DR signals in a distributed manner, thus giving autonomy and flexibility to customers. Both the solution based on contracts and the one based on parallel auction markets show abilities that go far beyond the capacity of typical aggregators.

With regard to the reliability and robustness of the solution, it is especially important to note that, from an architectonic standpoint, it is fully compliant with the architecture defined by the EI standard. Accordingly, the possible risks and points of failure that the solution may introduce must be sought in the auction markets, which, in order to ensure the network's reliability, are implemented using a conservative strategy. Specifically, when signals of type *simple* are sent, in order not to apply the ordered level, a node must purchase from other nodes the full gap of demand that this action produces; and when signals of type *delta* are used, the solution is implemented so that if, as a result of the execution of the algorithm, the amount of demand reduced does not cover the delta value, all agreements are canceled, the nodes being compelled to adopt a predefined level of consumption that ensures the reduction ordered by the system operator. In case of internal failure, the measures required for the protection of the network, as well as the mechanisms needed to shape a resilient infrastructure, are the same that any other node would require. As mentioned in Chapter 3 (Section 3.4), an ASPEM node only differs from other nodes in its behavior, which is actually a facet that the EI standard leaves open. Therefore, the procedure established by the system operator to replace or bypass a faulty node is expected to be the same for all the nodes.

As for the ability of the solution to scale, ASPEM nodes give rise to a hybrid overlay very similar to that which has been successful in P2P networks due to its ability to efficiently support a large number of clients. In the particular case of ASPEMs, these are expected to be implemented by technological companies which have to pass a certification process in order to operate in the electrical grid. Therefore, the conditions that the ASPEM must comply with, as well as its limits, are defined in a previous certification stage, thus combining reliability and scalability in a secure manner.

One of the most outstanding features of the architecture based on ASPEMs is that advanced types of signals can be handled (such as the *delta* one) keeping simple AMIs in the customers' facilities. In the simulations, customers, regardless of the type of signal sent by the system operator, always receive messages corresponding to the simplest profile of the OpenADR standard. This feature is inherited from the Agency Services model, and is possible because most of the complexity is taken over by ASPEMs. In the particular case of the Smart Grid, due to its huge dimensions, simpler devices in the customers' facilities means an important saving of financial and human resources, because simpler AMIs are cheaper and results in easier and less error-prone maintenance.

However, as experiments based on auctions show, adjusting the demand by using discrete levels of consumption brings new challenges. Specifically, when devices are ordered to adopt new consumption levels, fluctuations corresponding to the stabilization make it difficult to fully match the expected demand curve, making it possible that the output exceeds the expected value for short time periods. The system operator is responsible for minimizing this effect through a proper design of the market. Simulations reveal that two important design factors are: (*i*) the profile of the devices involved; and (*ii*) the duration of the event. The simulations show that the duration of the latter must be defined in accordance with the stabilization period of the former. When these parameters are not compatible, the system operator may decide not to authorize ASPEMs to run markets, so they would have to directly convey the incoming DR signals to the customers. In the experiments, this condition can be imposed by assigning values greater than zero to the priority of the OpenADR events.

As for the efficiency of the resultant demand curve, more advanced approaches are possible in creating outputs through discrete consumption levels. In the algorithms described in the Section 6.1.1, producers are selected iteratively so that the maximum available capacity of each of them is requested each time until the gap is covered. This means that, in a list of nodes, before requesting the *moderate* level of a node, if possible, the *high* and *critical* levels of the previous nodes are used. This approach, depending on the definition of the consumption levels, may lead to less efficient outputs. Therefore, with the aim of ensuring system efficiency, ASPEMs may be required to give preference to solutions based on specific set of levels or combinations of them. This condition can also be forced in auction markets. It must be noted that benefiting the overall system efficiency may surely not fit with the preferences contracted by the customers. For instance, a high priority user who is willing to adopt the *critical* level may be requested to only adopt the *moderate* level because the system considers it is a better solution. Accordingly, it is important to reach a compromise between overall efficiency and user rights.

According to the OpenADR standard, ASPEMs cannot communicate between themselves, since this is only possible between a VTN node and a VEN node that are directly linked. Although it is technologically feasible and easy to implement, this constraint eliminates the possibility that ASPEMs can be coordinated to form larger markets. In this regard, the most likely option would be to hold the market at a common VTN node, which would be provided and managed by the system operator. In this case, each ASPEM would deploy its broker agents in that node, or would communicate with it through a well-defined interface. In fact, the Agency Services model is intended to this type of solution: settings in which the Agency Services Providers deploy their broker agents in a remote business context. However, holding this type of market requires extra communication messages that, at the moment, are not supported by the OpenADR standard. However, it is important to study this approach in future research and check the advantages it may bring, as well as the new challenges it may pose.

The implementation of the simulation platform, which exclusively uses OpenADR messages for the communication between nodes, discloses some aspects of the standard for which the role of ASPEM offers advantages. The most important are:

- In the OpenADR standard, events are communicated by using the message "`OadrDistribitueEvent`". An important fact is that this message is not only used to communicate an event, but to define the status of the active events, so that if an event is not present anymore in the message is because it has been canceled. This approach frees VENs to manage the events' life cycle, but demands more capacity to the VTN nodes, such as the aggregators, which must know the specific events that are running in each VEN at all times. ASPEMs, due to their strong technological capabilities, successfully overcome this complexity.
- VTN can request information from its descending VENs by using the message "`OadrCreateReport`". ASPEMs, taking advantage of their capacity to process and store huge amounts of information, can cache or estimate information that is frequently requested, thus saving time and bandwidth resources.
- ASPEMs actually represent a virtual place in which VENs, through intelligent software agents, carry out peer-to-peer communications. That is, ASPEMs help in part to solve the restriction that prevents VENs from communicating directly. Markets and other types of exchange mechanisms are the confirmation of the advantages that bring this new virtual place.
- Since events are performed by many VENs in unison, at the start of the event, sudden drops may occur, as well as spikes at the end (as shown in the experiments). The solution proposed by the OpenADR standard is to define a window of time during which nodes can randomly choose to start and end their action. ASPEMs can enhance this mechanism by specifying the exact time in which each node has to start and end its activity to achieve a smooth transition.

Additional advantages would be possible if the OpenADR standard, as well as any other standard related to the Smart Grid, were designed considering the existence of supernodes. In particular:

- The profile *2.0a* of the OpenADR standard is only capable of applying signals of type *level*, so events that specify an amount of energy (*delta* signals) to be discarded are not supported. This decision aims to install very simple control devices in homes. Accepting signals of type *delta* would entail being able to translate the specified amount of energy into a specific operation point of the consumer's unit. In a scheme based on the Agency Services model, this task could be carried out by the ASPEMs, which would finally simply convey a combination of operation points to the local control devices. To make this possible, the profile *2.0a* should accept the definition of operation points.

- The capacity of sending and receiving messages corresponding to the *EiReport* service, which is intended to convey statistics and information about the units under control, is only available from the profile *2.0b*. However, if the capacity of sending this type of messages were also available in the profile *2.0a*, the control devices of the majority of users would be capable of sending important information to the ASPEMs, thus improving their planning and negotiation skills. It is important to remark that the only required capability is sending messages, and not be able to receive and process their content.

Regarding the security of communications, XMPP uses the *Transport Layer Security* (TLS) encryption protocol. Any further discussion about this topic is beyond the scope of this document.

– Politraumatizado, coma profundo, palidez, pulso filiforme, gran polipnea y cianosis. El hemitórax derecho no respira. Colapsado. Crepitación y angulación de la sexta costilla derecha. Macidez en la base pulmonar derecha con hipersonoridad en el ápex pulmonar. El coma se hace cada vez más profundo y se acentúa el síndrome de anemia aguda. Hay posibilidad de ruptura de arterias. ¿Alcanza? Yo lo dejaría en paz.
– A mí los enfermos se me mueren en la mesa.
"*Jacob y el otro*", Juan Carlos Onetti.

CHAPTER 7

Distribution of buyers in parallel auctions

The experiments of Chapter 6 reveal that parallel auctions, as a management system, can yield unsatisfactory results if buyers' participation is poorly distributed among available auctions. This fact represents an important risk, since, in practice, objective and particular factors actually lead the bidders to prefer some auctions over others, with the result that the overall participation usually ends up concentrated in a limited subset of auctions. In this scenario, most of the sellers and buyers fail to seal deals, thus eliminating one of the most promising, flexible, distributed mechanisms for managing environments such as computational grids and the Smart Grid. This risk, however, has been barely mentioned in the literature and no solution has yet been implemented to address its negative consequences.

This chapter, aware of the importance of finding a solution to the above problem, presents a mechanism intended to achieve a uniform distribution of buyers. However, as will be described, the introduction of a solution like this poses new challenges, amongst which are: respecting the distributed nature of parallel auctions, managing thousands of agents simultaneously, and preserving the essence of markets so that the motivation for bidding remains. Specifically, this chapter serves to:

- Explain how, in practice, objective and particular factors motivate buyers to prefer some auctions over others, and how this behavior can affect the overall performance of parallel auctions as a management system.
- Define the set of rules which would insure that the introduction of any mechanism intended to redistribute buyers' participation would maintain the players' motivation.
- Design and implement a scalable, reactive and non-blocking control mechanism that uniformly distributes buyers across the whole spectrum of available auctions. Furthermore, the mechanism is designed to deal with usual characteristics of auctions, such as the presence of starting prices and bids expressed in form of linear piece-wise functions.
- Evaluate through realistic experiments both the effect that the agglomeration of buyers actually has on the performance of management systems based on parallel auctions, and the effectiveness of the proposed control method to counteract it.

7.1 Lack of distribution in parallel auctions

In general, each buyer [1] that reaches the starting price of an auction will receive an invitation to participate on it, since sellers are interested in attracting as many bidders as possible in order to increase the competitiveness and achieve better prices. In large environments, such as the Smart Grid or computational grids, a buyer may receive hundreds or thousands of invitations. When there are much more buyers than sellers, the preferable situation is that buyers' participation is spread throughout all auctions, so that all sellers receive enough bids to cover their offers. However, a uniform distribution of the bids cannot be expected in settings inhabited by independent, autonomous and self-interested buyers. At this point it should be noted that, in many cases, buyers cannot participate in all auctions at the same time, since this would imply overbooking their bidding capacity, which is an action full of uncertainty that may entail an unaffordable cost for the buyer, or that may carry

[1] For the sake of simplicity, this document assumes that the roles of *buyer* and *bidder* are equivalent, and that likewise the roles of *seller* and *auctioneer* are equivalent.

high penalties for the system. As a result, in most of the scenarios, buyers have to limit their participation to their current bidding capacity at the moment.

In addition, buyers, as autonomous entities, cannot be forced to randomly choose their participation. In practice, unbiased shared factors may drive them to prefer some auctions to others, thus resulting that bids end up concentrated in a limited group of auctions. To the best of our knowledge, this condition has only been reported in [Hop08]. Specifically, during the experimental evaluation, the authors point out that the performance of their work may have been affected by the distribution of the buyers' participation, which in that case is performed at random. In general, the agglomeration effect is an issue related to the global functioning of the system, so that it does not enter into the scope of research projects principally focused on studying and evaluating the efficiency and scalability of specific algorithms. Actually, for this effect to be visible, it is necessary to: (*i*) simulate the entire parallel auctions market; and (*ii*) program software agents to follow intelligent strategies.

In the most general case, the information that buyers have to decide the distribution of their participation is the amount they are willing to buy, their internal valuation of the product, the amount auctioned by each seller, and the starting prices of the auctions. As will be shown, even this seemingly neutral information may encourage buyers to prefer some auctions to others. In any case, there are usually further context-specific factors that enhance the tendency to prefer some sellers to others. Some examples that demonstrate the power of common and particular factors to group buyers' participation are:

- Active auctions can be canceled in situations in which sellers have to sell their entire offer in order to clear the accepted bids. In this case, sellers that offer large amounts of the product are more likely to cancel their auction. For instance, this is the case of DR programs simulated in the Chapter 6, in which consumers can only adopt higher levels of demand when there are producers that supply them (that are willing to buy that new gap of demand). If the demand cannot be completely supplied, the consumer is not allowed to pass to the next level. This fact can lead the buyers abandoning the largest auctions, in order not to lose the amount bidden in an auction that has more probabilities of being canceled. Computational grids can live similar experience when a node needs a minimum set of computing resources to carry out a task. In general, the cancellation of auctions presents a real risk for buyers when overbooking is limited or disallowed.

- The rational behavior is that buyers try to bid as many units as possible. The best allocation of the available units of a buyer can be solved by means of an optimization algorithm (or a simplified version of it) that uses as input parameters the quantity auctioned by each seller and the bidding capacity of the buyer. In this regard, the more the correlation between the capacities of buyers, the greater the tendency to select the same auctions.
- Particular factors related to the players and products, such as their reputation and quality, are clear incentives for the buyers. Although buyers have the right to sustain their decision on these parameters, it is also true that an excessive zeal in these factors may not be justified and may turn out to be prejudicial for both players and the system. For instance, on Internet sites where the owner charges a commission for each transaction, this condition would mean important losses, whereas on fields such as the Smart Grid and computational grids it would lead to non-functional systems. Therefore, in order to ensure a minimum return system, imposing limits on the choices based on these factors is still required.

The excessive concentration of buyers causes the overall system efficiency to fall dramatically. In environments where parallel auctions are the most suitable solution, this behavior results in a huge waste of resources and leads to inoperative systems. Thus, many environments could benefit from a mechanism that helps to alleviate the effects of shared strategies. However, the solution has to be defined so that players remain motivated and the distributed nature of the process is preserved.

7.2 HUDP distribution mechanism

7.2.1 Basic description

When auctions are used, as a first step, sellers and buyers must register with a yellow-pages directory that enables them to contact each other [FIP03a]. Through this, agents can share when they work as sellers and buyers, and likewise what products they are willing to sell and buy. In addition, information related to each particular case can be provided, such as measures of quality and availability. An equivalent functionality can be achieved by using tables. More to the point, all sellers can register in a table that subsequently is queried by the buyers to find the auctions they can participate in. This approach is valid as long as the mechanism can give support for simultaneous access and can be adapted to distributed contexts, which implies facilitating the easy replication of its data and functionalities.

The power of the fact that buyers have to ask a controlled mechanism for the available auctions is that the set of auctions that are returned to each buyer can be filtered according to some rules. On the basis of this idea, this research proposes a mechanism suitable for distributed contexts by means of which subsets of auctions are assigned to buyers in accordance with a uniform distribution. In particular, to follow a uniform distribution guarantees that all buyers have the same probability of accessing a particular seller. It must be noted that striving to do a more intelligent assignation may lead us into the trap of trying to solve part of the clearing process in a centralized manner, when parallel auctions are intended to be the contrary.

The proposed solution is inspired by the functioning of hash tables. Actually, this is a hash table that: (*i*) returns subsets of sellers (or auctions) to buyers in accordance with a uniform distribution; and (*ii*) is accessible through an interoperation layer. This work will refer to this mechanism as *HUDP*, which stands for *Hash-based Uniform Distribution of Players*. The basic behavior of HUDP can be summarized in two steps (7.1):

i. Buyers register with HUDP. As a result, each buyer gets a unique anonymous token.
ii. A buyer, identified by his/her unique anonymous token, asks HUDP for the list of auctions in which he/she can participate. HUDP compiles the list of auctions that corresponds to the buyer and returns it to him/her.

7.2.2 Prerogatives and requisites

Including a mechanism as proposed may affect the incentives that lead players to participate, and can also alter the distributed nature of parallel auctions. Accordingly, the aim of this section is to define rules and conditions that ensure that the scenarios using HUDP remain competitive and feasible.

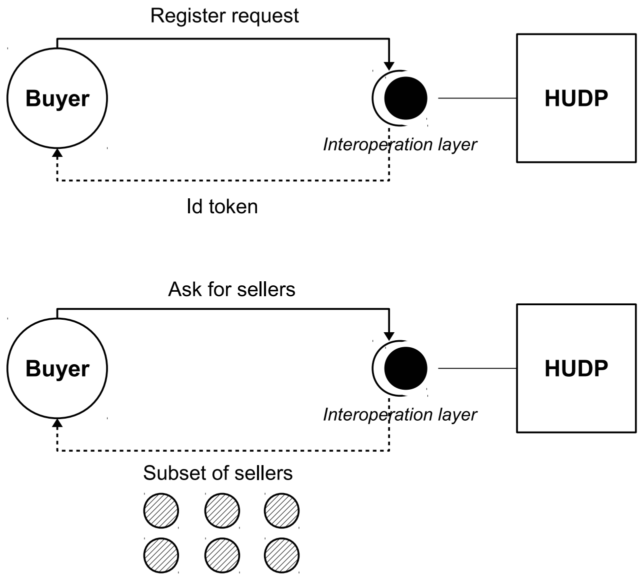

Figure 7.1: Overview of the HUDP mechanism.

Returning a subset of the original list of auctions to the buyers alters the initial condition of the market: buyers have fewer options to distribute their bids, and sellers have fewer participants in their auctions. In order to keep the essence of the market, it is necessary to limit the effect of this deformation and ensure some essential rights of the players. To this end, this work proposes to protect four basic prerogatives of players:

1. If in the original setting a buyer can receive enough invitations to allocate all his/her bidding capacity, then the mechanism must preserve this condition.
2. If in the original setting a seller can receive enough bids to sell all his/her offer, then the mechanism must preserve this condition.
3. The mechanism cannot be in detrimental to a specific player consciously.
4. The order in which buyers access the mechanism cannot affect the result of the queries.

In massive environments, such as the Smart Grid or computational grids, a mechanism like the one proposed would be accessed by hundreds or thousands of simultaneous players, so the implementation must be reactive, scalable, fail-safe and suitable for parallel processing. Furthermore, in order to preserve the distributed nature of parallel auctions, the solution cannot be based on a single entry point. According to all these conditions, the implementation must have the following features:

- *Non-blocking access*: Processing the query of an agent cannot block the access to other agents. In that case, thousands of agents may be waiting for their turn, thus causing starvation problems to the system and players, and extending the process in excess. This circumstance is emphasized in large distributed environments, where agents would need to receive a network message each time other agent finishes using the service. Apart from stressing the communication system, this approach would be very inefficient.
- *Replication*: Many separated instances of HUDP are necessary in order to avoid the risk entailed by the failure of single entry point. Moreover, the capacity to replicate the structure guarantees the scalability when required.
- *Reactivity*: The action of obtaining the list of the auctions available to a buyer cannot be time-consuming. Like in the classical implementation, the complexity of auctions must remain focused on taking decisions such as who are the winners of the auction (on the sellers' side), or which bid should be sent next (on the buyers' side). This is not desirable that the new mechanism adds new demanding tasks.

Actually, ordinary hash tables meet all the above-mentioned conditions, which is why the solution proposed in this Chapter is inspired by them.

7.2.3 Functionalities and interactions

The basic functionalities that must provide the interoperation layer of HUDP are:

- `RegisterPlayer`: This method enables buyers and sellers to register with the mechanism. Sellers register offers together with important information related to the auction, such as the starting price or the reserve price (if any). Depending on the conditions of the situation, buyers may only be required to provide their names, or additional fields such as their bidding capacity and the maximum price they are willing to pay. As a result of calling this service, each player receives a unique identification token. It is important to remark that the information required by this method is not used to determine which sellers should correspond to which buyers; instead, as will be explained, this is collected only for the purpose of obtaining statistics and global parameters that help to achieve a better uniform distribution of sellers and buyers.

- `GetSellers`: Each buyer agent uses this method to obtain the list of auctions in which he/she can participate. The buyer uses the identification token obtained in the previous method as input parameter. Based on this, HUDP finds the list of auctions that corresponds to the buyer. In most of the cases, this is a subset of all possible auctions.
- `RegistrationClosed`: A central authority uses this method to inform HUDP that the registration process is closed. Only the registered players at this point can participate in the new auctions. In practice, as will be explained in Section 7.3, providing a method like this is critical for the system performance because most of the operations used to achieve a uniform distribution can be carried out immediately after all players have been registered and before the auction starts, thus avoiding them having to be performed during the execution of the `GetSellers` method.
- `AuctionStarted`: A central authority uses this method to inform HUDP that a new auction period has started. This method is optional because HUDP may implicitly consider that the auction period starts immediately after the registration process is closed.

The natural sequence of calls is depicted in Figure 7.2. The steps are:

1. Buyers and sellers that want to participate in the auctions of the context, register with the HUDP by using the method `RegisterPlayer`. As a result, an identification token is assigned to each of them.
2. A central authority indicates that the registration phase is over by using the method `RegistrationClosed`. New players are not accepted for the coming auction period.
3. When HUDP ends processing the `RegistrationClosed` event, the central authority communicates that the auction period has started by using the message `AuctionStarted`.

4. Buyers use the method `GetSellers` to know which set of auctions is available to each of them. From this point, the auction runs as usual: buyers are free to decide in which auctions of the subset they participate.

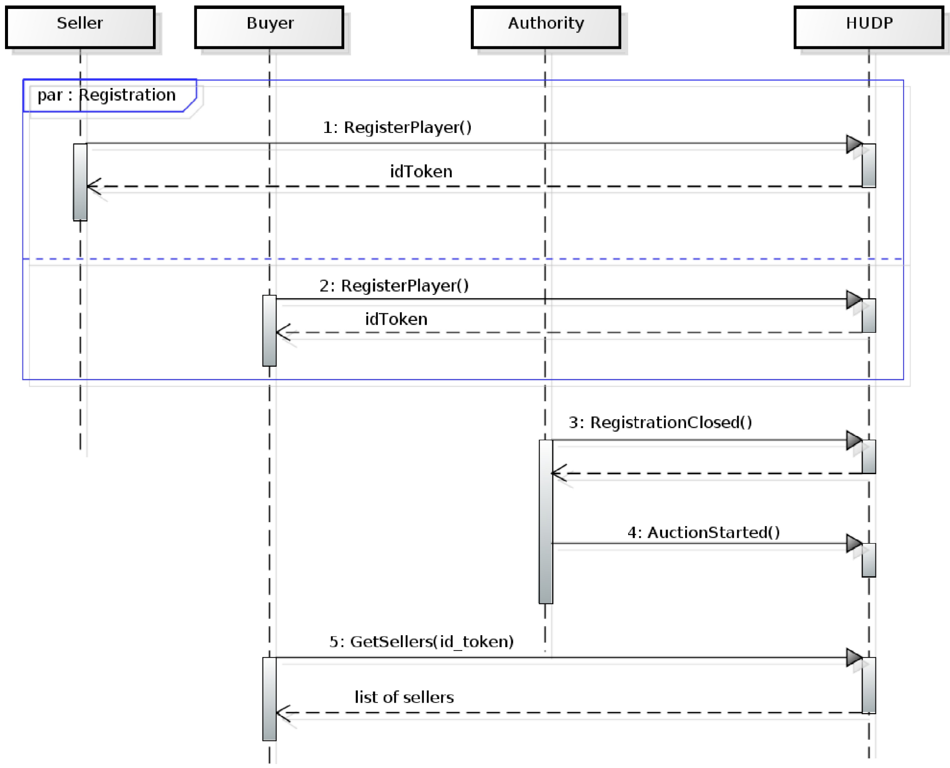

Figure 7.2: UML2 sequence diagram of HUDP.

The four methods proposed for the implementation of HUDP can be encapsulated by one or more agents of the platform, so that the information can be accessed by messages typical of software agents. In case they are provided as web services, a solution like the one described in [GLMS07] can be implemented.

7.2.4 Coexisting with standards

On agent-based solutions, the installation of yellow-pages directories is common. These are intended to providing current information about other agents registered in the platform: how to contact them, the capabilities they have, and the ontologies they use. In FIPA standards [FIP96], this role is played by agents of the platform that are called *Directory Facilitator* (DF) [FIP03a].

If auctions are run on a standard platform, the usual first step of agents is to register with the DF in order that sellers and buyers can find each other. In a scheme like this, including a mechanism such as HUDP would involve adding one more registration tasks. To avoid this extra step, two approaches are possible:

a) Including the functionalities of HUDP in those agents that play the role of DF. In essence, the information services provided by DFs are similar to that provided by HUDP: both are set of services whereby agents can register themselves and find other agents.

b) Acquiring the data needed by HUDP from the DFs. In this case, the extra step is carried out after the registration stage has been closed, but before auctions start.

The former approach is the most suitable one because:

- Agents that provide the functionalities of HUDP do not need to build an extra structure. The data needed to provide its functionalities can be obtained directly from the DF without exchanging extra messages.
- DF agents are usually deployed by the owner of the platform. According to the needs of the context, the owner can plan the necessary number of DFs to provide reliable services for the auctions.
- DF agents are well known by the platform agents, so accessing the services of HUDP does not require an extra effort.

7.3 Implementation

The aim of HUDP is to spread, whenever possible, the bidders' participation along the full range of auctions. This work proposes a simple implementation that achieves this goal, protects the prerogatives of the players, and fulfills the conditions laid down for implementation in the previous section. It is important to note that the performance of any solution devoted to this task, and thus the performance of the proposed implementation below, is conditioned by the following factors: (*i*) the bidding capacity; (*ii*) the number of buyers; and (*iii*) the uniform distribution of the capacity between buyers. The higher the value of these factors, the greater is the capacity of the solution to achieve better results.

The core points of the algorithm are:

i. Creating a uniform distributed list of sellers.

ii. Associating the identification token of each buyer to a start position in the list of sellers.

iii. Starting to collect sellers from that position until the *Prerogative 1* is fulfilled.

As a result of this basic mechanism (Figure 7.3), overlapped sets of sellers are assigned to the buyers. However, as will be shown, the need to ensure *Prerogative 2* requires a more advanced solution.

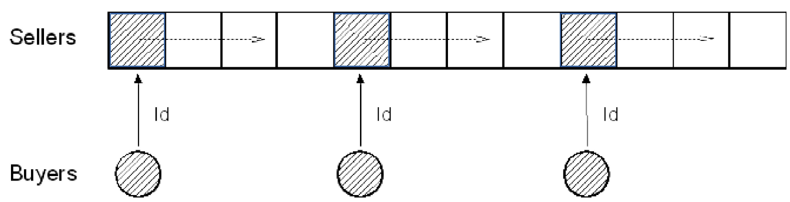

Figure 7.3: Buyers associated with a position of the list of sellers according to their identification token.

The operations of the distribution algorithm can be performed over two possible stages:

1. *Registration*: When the method `RegistrationClosed` is called. When processing this event, the agent (or the piece of software) that implements HUDP can extract valuable information from all registered buyers and sellers.
2. *Querying*: When the call to the method `GetSellers` is being processed. This is the period of time that takes to give to the buyer (who calls the method) the list of auctions in which he/she can participate.

In general, all operations performed during the *Registration* stage become saved processing time during the *Querying* one, which is the most important stage to shorten, since this means faster accesses for buyers.

In practice, the ratio of bidding capacity to amount offered, the ratio of number of buyers to number of sellers, and the existence of factors such as starting prices and many bidding levels, set scenarios of different complexity from the point of view of HUDP. The following sections put forward solutions for all these settings.

7.3.1 Common starting price

All auctions of the scenarios discussed in this section share a common starting price or do not have any at all.

7.3.1.1 Bidding capacity is greater than the amount auctioned

In this setting, when a buyer requests the list of sellers in which she can participate, the HUDP finds out the starting position that corresponds to that buyer, and starts collecting auctions from there until the bidding capacity of the buyer, plus an extra amount, can be fully allocated. The purpose of adding an extra amount of auctions is that buyers have the opportunity to run their own selection process on the set of auctions returned.

The challenge of this algorithm is to associate a starting position to each buyer so each seller receives enough bids to cover her offer (*Prerogative 2*). To achieve this, the minimum number of buyers needed per seller ($nBpS$) is calculated by using the average amount offered per auction (U_s), the average capacity per bid (U_b), and a constant value (C_f).

$$nBpS = \left\lceil \frac{\overline{U_s}}{\overline{U_b}} \right\rceil + C_f \tag{7.1}$$

Where $\lceil X \rceil$ is the ceiling function which gives the smallest integer that is greater than or equal to X. When the C_f constant is zero, $nBpS$ is equal to the average number of buyers needed to cover a typical auction. This number of buyers can be insufficient to cover the largest auctions. The C_f constant is necessary to counteract this condition. Furthermore, it increases competitiveness on the buyers' side.

In order to find out whether there are enough buyers to cover all sellers, the parameter α is used, which is calculated by using the number of buyers (nB), the number of sellers (nS), and the value $nBpS$. Specifically, when:

- α is smaller than 1, the number of bidders is not enough to cover all auctions.
- α is equal to 1, the number of bidders is enough to cover all auctions.
- α greater than 1, besides the number of bidders is enough to cover all auctions, the value of $nBpS$ can be increased to $nBpS_c$.

$$\alpha = \left\lfloor \frac{nB/nBpS}{nS} \right\rfloor \tag{7.2}$$

$$nBpS_c = \left\lfloor \frac{nB}{nS} \right\rfloor \tag{7.3}$$

Where $\lfloor X \rfloor$ is the floor function which gives the greatest integer that is smaller than or equal to X.

If the identification tokens of buyers are implemented by using a sequence of natural numbers, when α is equal to or greater than 1, the starting position (*pos*) of a buyer can be calculated by a simply division. It is necessary to note that, as a result of not using decimal points in the expressions, there are buyers whose position turns out to be greater than the maximum available. In this work, they are sequentially allocated to the auctions with the larger amount offered.

$$pos = \left\lfloor \frac{idBuyer}{nBpS} \right\rfloor \tag{7.4}$$

The only value calculated during the *Querying* stage is the starting position of the buyer, which is a very simple operation. The rest of the operations are done at the *Registration* stage. Once the starting position is known, the Algorithm 1 is used to pick the auctions corresponding to the buyer.

```
bc ← bidding capacity of the buyer;
startPos ← starting position in the list of registered auctions;
pos ← startPos;
selectedAuctions ← empty list of auctions;
while (bc > 0) and (pos <= maxPosition) do
    auction ← auction at the position pos of the list;
    add auction to selectedAuctions;
    bc ← bc − (amount offered in auction);
    pos ← pos +1;
end
pos ← 1;
while (bc > 0) and (pos < startPos) do
    auction ← auction at the position pos of the list;
    add auction to selectedAuctions;
    bc ← bc − (amount offered in auction);
    pos ← pos +1;
end
return selectedAuctions;
```

Algorithm 1: Get the list of auctions when there is common starting price and the bidding capacity is greater than the amount auctioned.

On the buyers' side, the competition can be increased by artificially raising the bidding capacity of the buyer in the above algorithm. This causes that more auctions than necessary are assigned to each buyer, thus allowing them to run their own selection over the returned list of auctions. In addition, the constant C_f can be used to manually increase the number of buyers per seller, allowing the latter to select from more buyers.

7.3.1.2 **Bidding capacity is smaller than the amount auctioned**

In principle, lack of bidding capacity is not the ideal setting for any auction. Under this condition, since there is not enough bidding capacity to cover all auctions, the only possible improvement is to reduce the negative impact that may cause the over-concentration of bids. To this end, with regard to the approach followed in the above case, instead of assigning many buyers per seller, the algorithm assigns groups of buyers to groups of sellers [1]. In this manner, if buyers share preferences when selecting auctions, the clustering effect is limited to the enclosing group of sellers corresponding to each buyer (Figure 7.4).

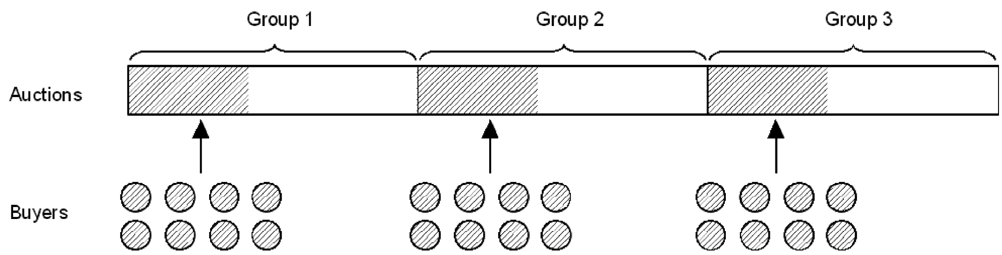

Figure 7.4: Over-concentration of buyers is limited by the frontier of each group of sellers.

This type of scenario is detected at the end of the *Registration* stage, when the value of α turns out to be smaller than 1. The maximum number of groups of sellers (nS_g) that can be created depends on the minimum number of buyers per auction. In particular, new groups of sellers can be defined as long as there are enough buyers to cover at least one auction of the group. It must be noted that for each group the ratio of bidding capacity to amount offered is maintained respect to the original setting. The more groups of sellers created, the smaller the effect of strategies, but also the less the competition among both sellers and buyers. Once the number of groups of sellers is chosen, the number of sellers per group ($nSpG$) is calculated in order to obtain the starting position of each buyer.

$$nS_g = \left\lfloor \frac{nB}{nBpS} \right\rfloor \tag{7.5}$$

$$nSpG = \left\lceil \frac{nS}{nS_g} \right\rceil \tag{7.6}$$

$$pos = \left\lfloor \frac{idBuyer}{nBpS} \right\rfloor * nSpG \tag{7.7}$$

[1]This is a generalization of the case when α is equal to or greater than 1, for which the size of the group of sellers is 1.

The algorithm that builds the list of auctions corresponding to each buyer (Algorithm 2) is slightly different from that used when α is equal to or greater than 1. Now every buyer must include, at a minimum, all the sellers of the group he has been assigned. As in the previous setting, all operations, except that related to the calculation of the position, are carried out at the end of the *Registration* stage, thus guaranteeing the fast access of buyers.

```
bc ← bidding capacity of the buyer;
startPos ← starting position in the list of registered auctions;
pos ← startPos;
nSpG ← number of sellers per group;
n ← 0;
selectedAuctions ← empty list of auctions;
while (bc > 0) or (n < nSpG) do
    auction ← auction at the position pos of the list;
    add auction to selectedAuctions;
    bc ← bc – (amount offered in auction);
    pos ← pos +1;
    n ← n +1;
end
return selectedAuctions;
```

Algorithm 2: Get the list of auctions when there is common starting price and the bidding capacity is smaller than the amount auctioned.

7.3.2 Starting price and bidding levels

This section deals with auctions where each seller can define a starting price and offers multiple units of the product. In these contexts, buyers usually make bids with linear piece-wise functions [SS01, DJ03] like the depicted in Figure 7.5. All these features entail an extra level of complexity. Actually, a precise uniform distribution of bids cannot be defined because: (*i*) the existence of starting prices means not all bids are valid for all auctions; (*ii*) only some sections of a bid (of the piece–wise function) may be applied to an auction; and (*iii*) the number of sections per bid may vary between buyers. Below, an approach based on the previous solution that overcomes many of these difficulties is put forward.

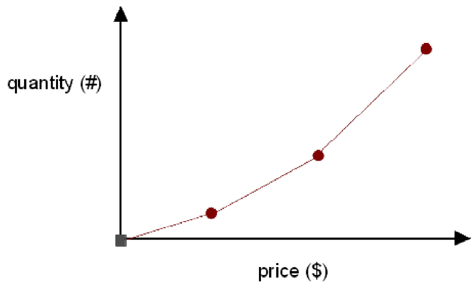

Figure 7.5: Linear piece-wise function.

7.3.2.1 Bidding capacity is greater than the amount auctioned

In this case, the parameter $nBpS$ is calculated as the mean of the number of buyers needed for each auction ($nBpS_i$). The parameter α is calculated as in the Expression 7.2.

$$nBpS = \frac{\sum_i nBpS_i}{nS} \tag{7.8}$$

As a first step, auctions are sorted by the starting price in ascending order. Next, each buyer is initially associated to the first auction in which she can participate. This condition is determined by both the starting price of the auction and the price of the lower level of the buyer's bid (the first section of the linear piece-wise function). Since the list of auctions is ordered, at least one section of the bid is valid for each auction from that position. Furthermore, in order to facilitate subsequent operations, each auction keeps a list of the buyers that start at that position (henceforth *buyers list* of the auction). Figure 7.6 illustrates an example in which three auctions are ordered according to the starting price. Each auction holds a list of the buyers that are initially assigned to that position of the list of auctions. Since the list is ordered by the starting price, all buyers assigned to the auction A can also participate in the auctions B and C. Likewise all buyers assigned to the auction B can also participate in the auction C.

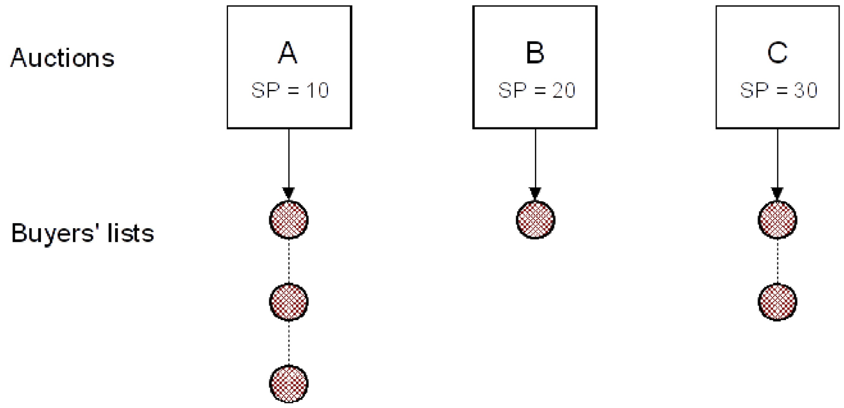

Figure 7.6: Buyers lists of the auctions.

Once the mechanism knows which buyers start at each auction, it calculates the number of surplus buyers per auction with respect to the parameter $nBpS$. Moreover, the number of surplus buyers is accumulated along the ordered list of auctions. Figure 7.7 illustrates an example in which $nBpS$ is 2. In the auction G there are two extra bidders accumulated. Note that in the auction F the number of surplus buyers is decremented because it has fewer bidders than necessary (than $nBpS$). The mechanism knows the position of each buyer in the *buyers list* that holds each auction, so that all those buyers whose position is greater than $nBpS$ are classified as surplus elements. All this information is obtained at the end of the *Registration* stage.

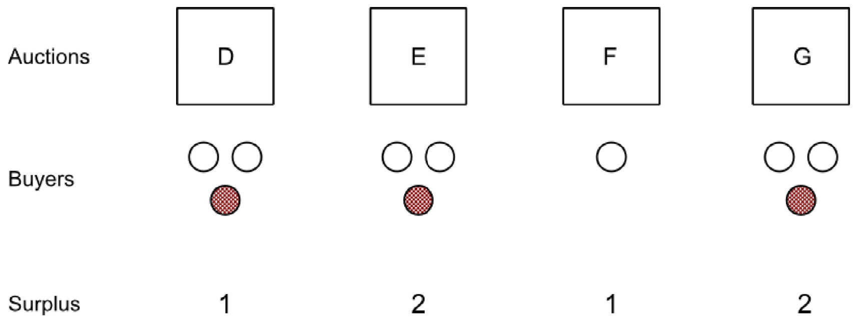

Figure 7.7: Example of extra buyers accumulated over the auctions.

Next, the starting position of a buyer is modified according to the surplus elements. Specifically, if the buyer has been qualified as a surplus element in the auction at which she has been initially assigned, her starting position is moved forward until taking up an auction requiring more bidders. Figure 7.8 illustrates a different example where $nBpS$ is 2. The surplus elements of the auctions H and I are re-assigned to the auctions J and K. By shifting the starting position of the buyers, all auctions receive help to find the minimum number of bidders defined by $nBpS$. In practice, bids with high starting prices are populated with elements initially assigned to auctions with lower starting prices.

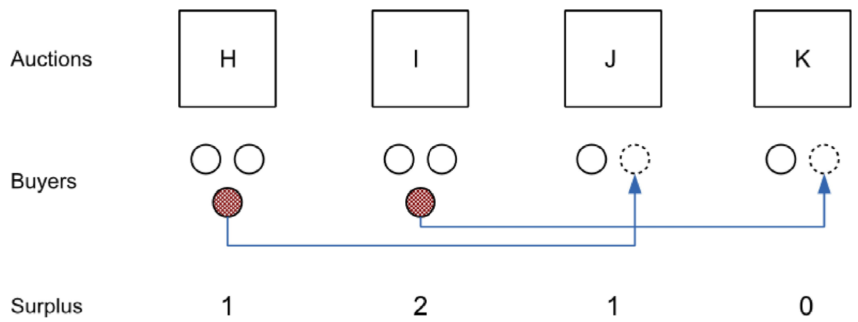

Figure 7.8: Example of shifting the starting position of buyers according to the surplus elements.

The correction of the position is performed at the *Querying* stage, when the corresponding buyer asks for her list of auctions. The operation is simple and fast because all data structures are built at the end of the *Registration* stage. Furthermore, it must be noted that the process must be performed for each bidding level (for each section of the linear piece-wise function).

In addition, in order to reduce the alteration that the existence of many bidding levels creates, the bidding capacity assigned to an auction is accumulated along the bidding levels. Therefore, an auction is only considered in the next levels of the loop if the accumulated bidding capacity (from the previous bidding levels) does not yet cover all the amount auctioned. The Algorithm 3 describes the operations needed to obtain the list of auctions in this setting.

```
selectedAuctions ← empty list of auctions;
foreach bidding level of the buyer do
    bc ← bidding capacity of the buyer at this level;
    firstAuction ← initial auction assigned to the buyer;
    firstAuctionPos ← position of firstAuction in the list of
      auctions;
    buyerPos ← position of the buyer in the buyers list of
      firstAuction;
    if buyerPos <= nBpS then
        startPos ← firstAuctionPos;
    else
        startPos ← find the position of the next auction that needs
          bidders;
    end
    pos ← startPos;
    while (bc > 0) and (pos <= nS) do
        auction ← auction at the position pos of the list;
        add auction to selectedAuctions;
        bc ← bc − (amount offered in auction);
        pos ← pos +1;
    end
    pos ← firstAuctionPos;
    while (bc > 0) and (pos < startPos) do
        auction ← auction at the position pos of the list;
        add auction to selectedAuctions;
        bc ← bc − (amount offered in auction);
        pos ← pos +1;
    end
end
return selectedAuctions;
```

Algorithm 3: Get the list of auctions when there are multiple starting prices and the bidding capacity is greater than the amount auctioned.

7.3.2.2 Bidding capacity is smaller than the amount auctioned

As when the starting price is common to all players, the best that can be achieved in this setting is to assign groups of buyers to groups of sellers. However, since the set of accessible auctions for each buyer is different, it is not possible to create static groups of auctions, but only an approximation.

When a buyer accesses HUDP, the algorithm first extracts the list of all auctions that are accessible to him. Also, according to the information collected at the *Registration* stage, the list of all possible buyers for that list of auctions is extracted. By using this information, an estimation of the number of groups of sellers is obtained (Expressions 7.5 and 7.6). The group that belongs to the buyer is calculated from his position in the *buyers list* of the auction. When this position is greater than $nBpS$, the buyer jumps to the next group of sellers. Furthermore, the accumulated surplus elements are considered to know how many times the buyer has to jump. The starting position is obtained by means of the Algorithm 4, and the list of auctions to be returned to each buyer is obtained through the Algorithm 5.

7.4 Experimental evaluation

Simulations are based on OpenADR events that last 90 minutes, which are managed using three consecutive markets of 30 minutes. As in Chapter 6, all simulations are run over the *IEEE 13*-node test feeder. Furthermore, henceforth, in order to provide more reliable results, all data shown for each scenario is the average of ten simulations.

As for strategies, in the absence of other context-specific factors, it is assumed that buyers strive to allocate their full bidding capacity in the auctions with smaller offers. This condition is desirable for buyers (producers) because sellers (consumers) who do not sell all their demand have to cancel the auction, in which case all bids sent to that node have to be dropped.

Table 7.1 summarizes the scenarios used in experiments. In addition to the amount of energy offered and bidden, the table includes two important factors in the understanding of the distribution mechanism and the results: (*a*) the ratio of bidding capacity to amount auctioned (henceforth R_{ca}), which decreases in each scenario; and (*b*) the ratio of number of buyers to number of sellers (henceforth

```
Function CalculateStartPos (list of auctions) is
    startPos ← position of the first auction available for the buyer;
    while startPos < nS do
        auction ← auction at the position startPos of the list;
        buyerPos ← buyerPos + position of the buyer in auction;
        buyerPos ← buyerPos + (number of surplus elements);
        if buyerPos < nBpS then
            break loop;
        end
        startPos ← startPos + nSpG;
    end
    return startPos;
end
```

Algorithm 4: Calculate the start position of a buyer when there are multiple starting prices and the bidding capacity is smaller than the amount auctioned.

R_{np}). The values shown for each scenario is the addition of the outcomes of the three markets. It is also necessary to mention that, in all simulations, the value of the constant C_f is 0 [1].

Table 7.2 shows the results for the *Scenario 1*, which stands out for having the bidding capacity much larger than the amount auctioned. If buyers do not follow a common strategy, the results are good even without using any special distribution mechanism, since, due to the large number of players, the participation thereof behaves as a uniform distribution. However, it must be noted that, although there is a large bidding capacity, there are a significant number of empty auctions (auctions that do not receive any offer). Likewise, there are a significant number of canceled auctions (auctions that do not receive enough buyers to allocate the entire offer). These results improve when HUDP comes into action. Actually, due to the better

[1]The effect of this constant is studied in detail later.

```
selectedAuctions ← empty list of auctions;
foreach bidding level of the buyer do
    bc ← bidding capacity of the buyer;
    startPos ← CalculateStartPos(list of auctions);
    nSpG ← number of sellers per group;
    n ← 0;
    while (bc > 0) or (n < nSpG) do
        auction ← auction at the position pos of the list;
        add auction to selectedAuctions;
        bc ← bc − (amount offered in auction);
        pos ← pos +1;
        n ← n +1;
    end
end
return selectedAuctions;
```

Algorithm 5: Get the list of auctions when there are multiple starting prices and the bidding capacity is smaller than the amount auctioned.

distribution of buyers, there are not empty auctions and the number of canceled auctions is halved. When buyers have incentives to use common strategies, HUDP practically avoids their effect. In contrast, when this mechanism is not used and players adopt common strategies, the performance falls dramatically.

As shown in Table 7.3, when using starting prices and multiple bidding levels, the distribution mechanism proposed in Section 7.3.2.1 achieves similar improvements. Furthermore, the effect of adopting common strategies is also largely avoided.

The *Scenario 6* represents the case in which the bidding capacity is not enough to cover the amount auctioned, and the number of buyers is smaller than the number of sellers. As explained in Section 7.3.1.2, here avoiding the effect of strategies can

Scenario	Offered (kW)	Bidden (kW)	R_{ca}	R_{np}
1	12653	36259	2.86	4.01
2	16211	40262	2.48	2.32
3	16937	31848	1.89	2.66
4	16753	26723	1.60	2.20
5	22789	25593	1.12	1.60
6	33602	25697	0.76	0.99

Table 7.1: Description of the scenarios used in the experiments.

	Without HUDP		With HUDP	
	Random	Strategy	Random	Strategy
Exchanged (kW)	10612	492	11672	11521
Offer covered	84%	4%	93%	91%
Canceled auctions	22	1	11	12
Empty auctions	22	245	0	0

Table 7.2: Scenario #1: Results when starting prices are not used.

	Without HUDP		With HUDP	
	Random	Strategy	Random	Strategy
Exchanged (kW)	8493	950	9949	8727
Offer covered	67%	7%	79%	69%
Canceled auctions	31	8	33	49
Empty auctions	41	222	1	4

Table 7.3: Scenario #1: Results when using starting prices and multiple bidding levels.

	Without HUDP		With HUDP	
	Random	**Strategy**	**Random**	**Strategy**
Exchanged (kW)	11663	487	11535	11565
Offer covered	45%	2%	45%	45%
Canceled auctions	249	3	285	164
Empty auctions	179	739	177	271

Table 7.4: Scenario #6: Results when not using starting prices.

	Without HUDP		With HUDP	
	Random	**Strategy**	**Random**	**Strategy**
Exchanged (kW)	10051	752	10824	10883
Offer covered	39%	3%	42%	42%
Canceled auctions	168	10	285	94
Empty auctions	314	715	177	431

Table 7.5: Scenario #6: Results when using starting prices and multiple bidding levels.

be only hoped for. Tables 7.5 and 7.4 show that strategies are successfully avoided, and also confirm that, when strategies are not present, the results are practically similar to those of the classical approach.

Figure 7.9 depicts the improvement achieved for each scenario. The general trend is that the improvement achieved by HUDP is reduced as the ratio R_{ca} decreases. This occurs because there are fewer options to achieve a better distribution. However, the relevant fact is that the effect of adopting strategies is largely avoided. In addition, another visible effect is that the introduction of HUDP does not worsen the result if players are not influenced by strategies. In general, HUDP helps to achieve a better balance between canceled auctions and empty auctions. Figure 7.9 also depicts that, when strategies are not used and the ratio R_{ca} is not high, not clear winner appears. It happens because HUDP has no room to work. In these settings, the outcome is primarily driven by the auctions that buyers select, which is a random process.

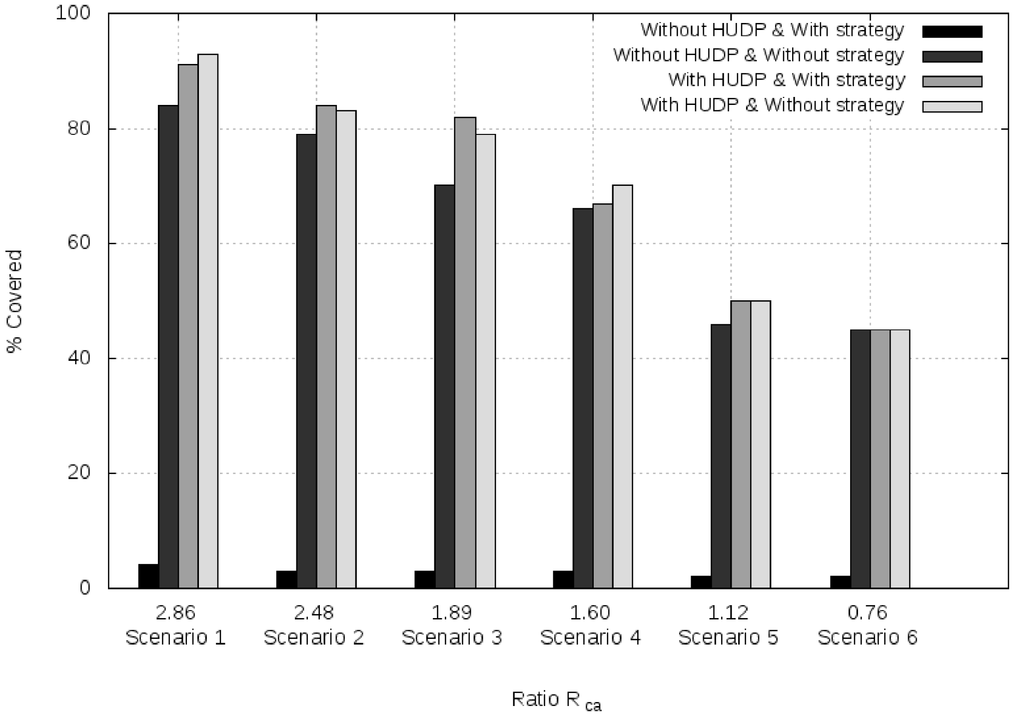

Figure 7.9: Histogram of the covered auctions in each scenario.

Figure 7.10 depicts the same data represented with lines of points. In this graph we can see that, in spite of the bigger difference between two ratios R_{ca} corresponds to the gap between *Scenario 2* (2.48) and *Scenario 3* (1.89), the improvement obtained between them is small. This is because the high value of the ratio R_{np} in the *Scenario 3*, which has a positive effect on the achievement of a uniform distribution.

Next, the effect of the constant C_f is studied more closely. The aim of C_f is to artificially increase the average number of buyers that are necessary to cover a typical auction. The motivation in doing so is that larger auctions receive enough bids to cover their offers, thus avoiding having to be canceled. However, another direct consequence of increasing the number of buyers per seller is that HUDP has less space to achieve a uniform distribution, thus increasing the number of empty

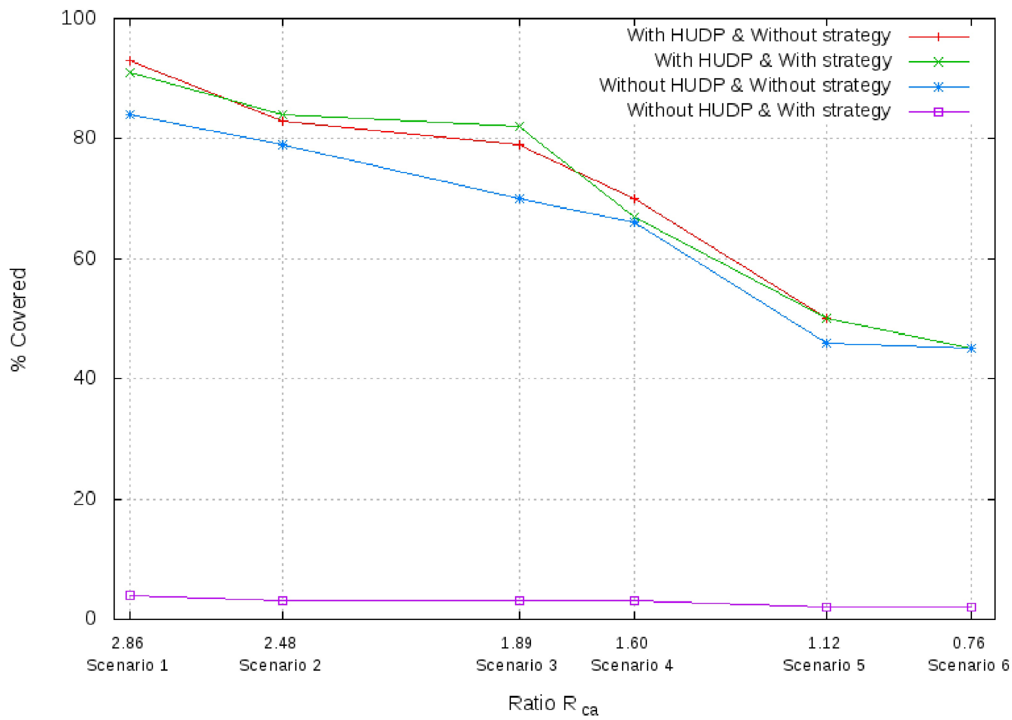

Figure 7.10: Percentage of covered auctions depending on the ratio R_{ca}.

auctions. From the analytical point of view, this condition is stated in Equation 7.1 (Section 7.3.1.1): increasing $nBpS$ through C_f may cause that the parameter α becomes smaller than 1, thus requiring HUDP to adopt a more ineffective distribution strategy (Section 7.3.1.2). Therefore, by definition, the ability to achieve better results through C_f is limited.

Figure 7.11 depicts the simulation of several scenarios with different values for the parameter R_{ca}. For each scenario, the value of the constant C_f varies from 0 to 10. In these cases, it is assumed that players do not adopt any strategy, so the auctions in which buyers participate are randomly chosen. The gray circle drawn in each line marks the point from which the value of α becomes smaller than 1. Figures 7.12 and 7.13 depict the evolution of the number of empty and canceled auctions with the increasing of C_f.

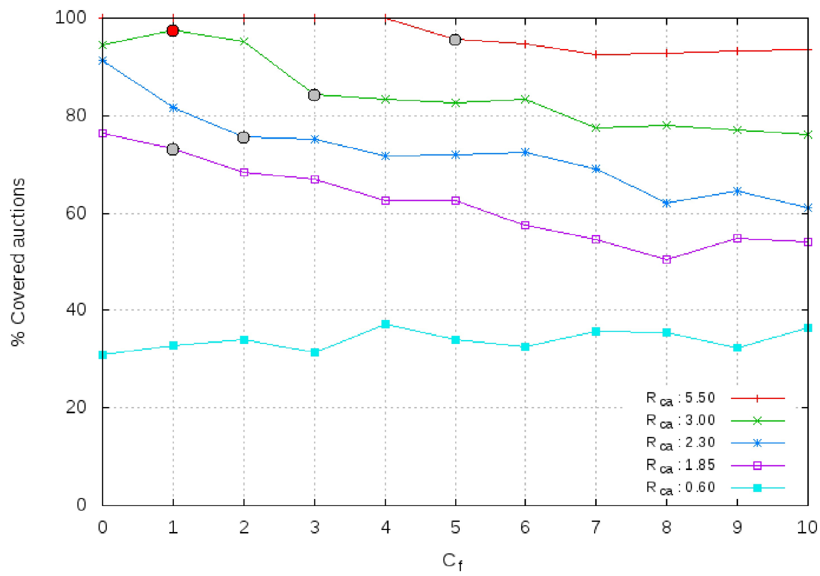

Figure 7.11: Effect of the constant C_f on the peformance of the HUDP mechanism.

It can be noted that the lines, despite following a general trend, have small fluctuations. This happens because buyers randomly select the auctions they participate in, so that the performance of the algorithm may vary when the differences between experiments are not significant. In particular, the general trend is that as

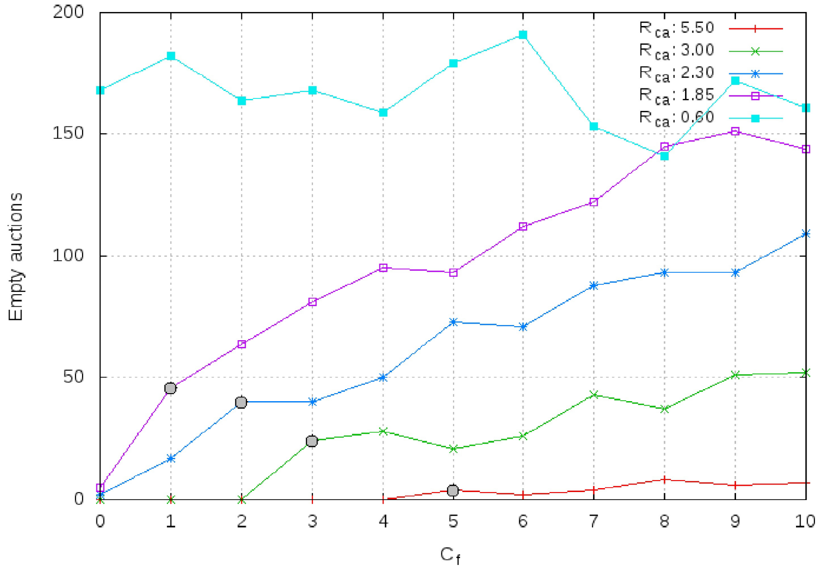

Figure 7.12: Evolution of the number of empty auctions with the increasing of C_f.

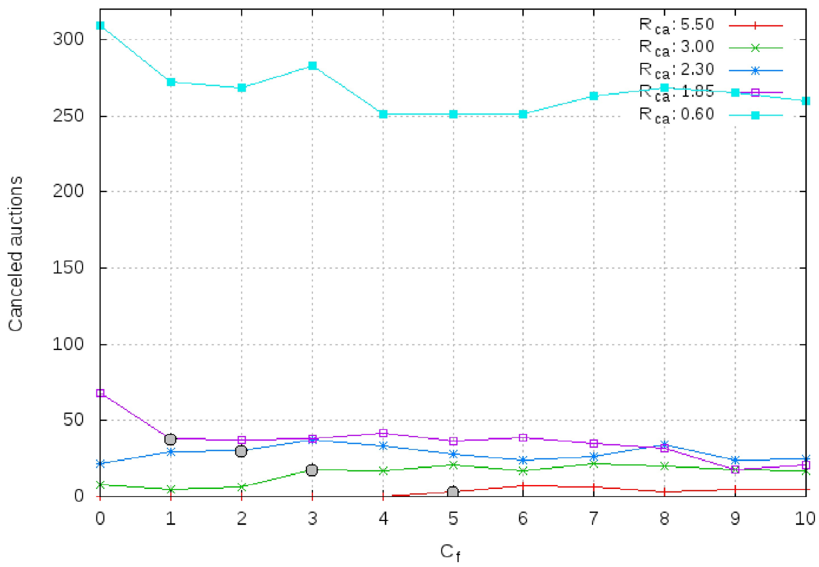

Figure 7.13: Evolution of the number of canceled auctions with the increasing of C_f.

C_f	**Offer covered**	**Empty**	**Canceled**
0	94.44%	0	8
1	97.55%	0	5
2	95.08%	0	6

Table 7.6: Performance of the scenario with ratio R_ca equal to 3 when the constant C_f varies.

the value of the C_f constant grows, there are more empty auctions and fewer canceled auctions. However, in this regard, Figure 7.13 depicts that the number of canceled auctions do not decrease continuously. This is due to the fact that when α becomes smaller than 1, HUDP starts to assign groups of buyers to groups of sellers (because having many buyers per seller is not possible anymore), so there is not guarantee that a specific auction receives enough bids to cover its offer. Thus, the power of C_f to reduce the number of canceled auctions is indeed limited. In general, by definition, as the value of C_f is increased, the performance tends to be more like to a solution without HUDP.

In Figure 7.11, the line corresponding to the scenario with R_{ca} equal to 3.00 illustrates an important fact: increasing the C_f constant can be beneficial because of a better balance between empty and canceled auctions. In the graphic, the point that achieves a better result is marked in red, and its data is shown in Table 7.6. Figure 7.11 also depicts that, as expected, the best balance is always achieved before the value of α becomes smaller than 1. Accordingly, the search space of the optimal value of C_f is limited to the range of points for which α is greater than 1, which simplifies the searching process.

The C_f constant also helps to decrease the number of canceled auctions in those cases in which α is initially smaller than 1. However, in this type of scenario, as discussed above, there is less chance that the extra units end up concentrated in the sellers that require them, so the effect of C_f is more limited. Conversely, the randomness of the selection process is more present in these cases. In fact, in the selected scenarios, when the number of canceled auctions decreases, the improvement in the overall performance does not exceed 10%.

Figure 7.14 depicts the same scenarios when agents follow shared strategies. It can be noted that, when α is greater than 1, increasing the constant C_f brings no benefit. This is the result expected because HUDP fights strategies by clustering the buyers; while a greater value of C_f leads to create fewer groups of buyers. As a matter of fact, if α is greater than 1, when the value of C_f increases, the graph shows that the performance always decreases rapidly. Conversely, when α is smaller than 1, as discussed, the effect of C_f is more limited because it is affected by the random selection of sellers by buyers.

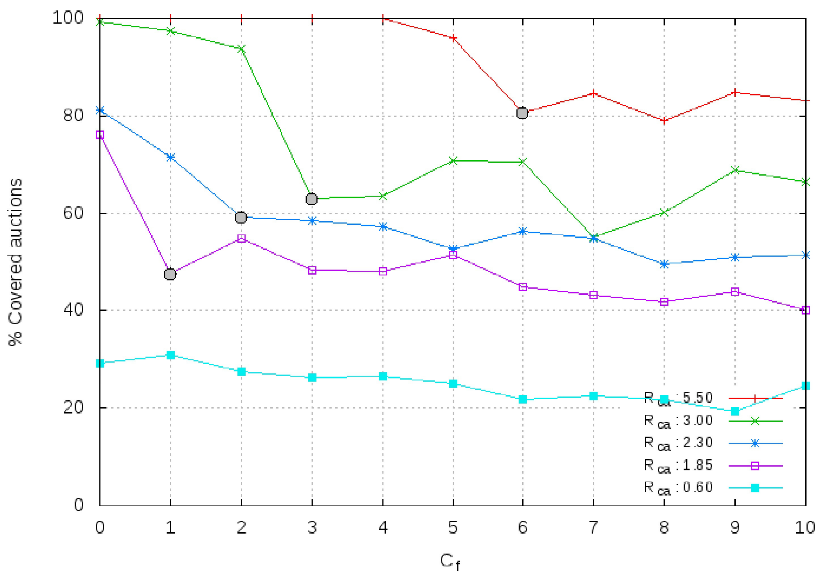

Figure 7.14: Effect of the constant C_f on the performance of the HUDP mechanism when players adopt shared strategies.

In conclusion, when buyers do not follow shared strategies, the performance of the HUDP mechanism can be finely tuned through the C_f constant. Specifically, a better balance among empty and canceled auctions can be achieved by increasing this parameter. Furthermore, the search space for finding the optimal value of C_f is bounded by the parameter α, which must be kept greater than 1. However, when buyers have incentives to adopt shared strategies and the parameter α is greater than 1, increasing C_f worsens the overall performance because buyers tend to be less clustered. When α is smaller than 1, regardless of the presence of strategies, the effect of C_f is limited because the random selection of sellers by buyers has more impact on the result.

7.5 Discussion

The proposed implementation only ensures the *Prerogative 2* at theoretical level. In practice, all those auctions that require more bidding capacity than the average determined by $nBpS$, run the risk of being canceled due to the lack of bids. As explained, the effect of this shortcoming can be reduced through the constant C_f, which helps to increase the level of competition on the buyers' side. However, the reader must keep in mind that, by default, the proposed algorithm also tries to alleviate this condition by allocating the surplus buyers to the more demanding auctions. If the scenario has auctions rather different from the average, more aggressive mechanisms are still possible, such as including, by default, the largest auctions into each group returned to each buyer. Although this measure would affect to the overall performance of the algorithm, it would ensure the *Prerogative 2* at practical level. On the contrary, some authorities could find preserving this shortcoming attractive because it works as an incentive for sellers to present auctions within average limits, thus contributing to the overall performance of the involved system.

Moreover, returning subsets of auctions to the buyers, as well as assigning subsets of buyers to auctions, results that some players have better opportunities than others. However, this always happens randomly, since the assignation is simply based on the anonymous token provided by the mechanism to the players. Furthermore, at the implementation level the tokens are provided randomly, independently of the registering order of the players. In any case, the use of HUDP, despite knowing that may randomly benefit some players, can be imposed by a central authority in order to improve the overall outcome of the allocation process, as well as to increase the opportunities of most of the players to reach a deal. This is the case of the Smart Grid, which is planned to provide an essential service to the society; given the existence of common strategies to some extent, the system operator can choose to improve the distribution of buyers and limit the negative effects of clusters.

On the other hand, HUDP works analogously to hash tables and therefore is able to process players' requests rapidly. As previously mentioned, most of the parameters used in the algorithms are calculated at the *Registration* stage. Moreover, the complexity class of the Algorithms 1 and 2 is $O(n)$, where n is the number of sellers. The complexity of the Algorithm 3, in which linear piece-wise functions are used to form the bids, is $O(k * n)$, where k is the number of segments of the bidding function. It also must be noted that, when the bidding capacity is smaller than the amount auctioned, n tends to be equal to the number of sellers per group ($nSpG$), which is expected to be significantly smaller than the number of sellers.

As mentioned in Section 7.2, HUDP is designed to be reactive, scalable and able to handle simultaneous accesses. In addition, advantages of installing HUDP are:

- *Fewer messages exchanged*: The number of messages used by the sellers to invite buyers is decreased significantly when using the HUDP, thus improving the overall performance of the communication system. The scale of the reduction is shown below.
- *More difficulties in adopting collusive behavior*: When using auctions there is always the risk that players collude in order to alter the prices in either direction. These options are reduced by HUDP, since specific players do not have the certainty of sharing the same space. Further, in the registration stage, by enforcing the use of anonymous tokens, HUDP has the ability to anonymize the players so that they are not able to recognize each other.
- *Early discard of unfeasible auctions*: Auctions that need to sell a very high amount of the product or have high starting prices may be impossible to cover even by collecting all available buyers. This type of auctions is detected and canceled early on HUDP, so that no buyer is assigned to them, thus improving the overall system efficiency.
- *Reduce the length of sequential processes*: In scenarios where multiple iterations are possible, there are proposals based on buyers that move across auc-

tions in order to find the most appropriate one to place their bids. On this basis, in [Str05] auctions are run throughout several stages, so that buyers can jump through auctions at each stage; and in [PS06] auctions are run sequentially so that buyers participate in auctions depending on their standing prices. Any of these approaches may turn out to be unfeasible in large contexts with thousands of parallel auctions. In these settings, HUDP would help to split buyers and sellers in groups, so the length of the sequential process is limited.

As for limitations of HUDP, it should be mentioned that:

- The current implementation of HUDP does not consider the presence of combinatorial auctions. To maintain the performance of HUDP in their presence, it is necessary to enrich the algorithms with techniques typical of this area [San02] and heuristic methods. The solution will depend on the types of combinatorial bids accepted by the system.
- HUDP has been specifically designed for systems in which parallel auctions are synchronized: they all are expected to start at the same time, while the period for negotiations is limited. However, this feature is typical of online contexts and management systems.
- According to the life cycle of HUDP, players can only participate in auctions if they have registered with the mechanism during the *Registration* stage (see Section 7.3), which is a step typically required in virtual environments and management systems.
- HUDP supports bids in form of linear piece-wise functions, which is one of the most common forms proposed in the literature [SS01, PJ05]. Other types of expression, such as continuous functions, would require modifying the algorithms.

Below, the reduction in the number of messages exchanged between buyers and sellers is shown. In the classic implementation, once a seller wants to start a new auction, it sends a message to all available buyers that are interested in buying the product. Thus, in a typical environment with parallel auctions, each buyer will normally receive as many requests as there are sellers. If invitations are implemented as FIPA interactions of the type *Request* [FIP02c] or *Query* [FIP02b], this step entails the following number of messages:

$$(nS * nB) + (2 * nB) \tag{7.9}$$

The first operand corresponds to the number of invitations sent by the sellers to the consumers; and the second corresponds to the `agree` and `inform` FIPA messages that are sent to the buyer to confirm the action.

By using HUDP, buyers get the list of available sellers from the table instead of receiving an explicit invitation message. Once all buyers have queried the table, if necessary, the agent implementing HUDP may be required to inform the sellers about the number of buyers that each of them should expect. Considering this, the maximum number of messages exchanged is reduced to:

$$(2 * nB) + nS \tag{7.10}$$

The first part belongs to the query carried out by the buyers and the corresponding reply from the HUDP's agent; while the second operand corresponds to the inform messages sent by the HUDP's agent to the sellers when required. As a matter of fact, the effect of decreasing the number of exchanged messages is noticeable during the simulations, which are shorter and faster when the distribution mechanism is used.

El hombre, mis hijos, es como un río. Tiene barranca y
orilla. Nace y desemboca en otros ríos. Alguna utilidad
debe prestar. Mal río es el que muere en un estero.
"*Hijo de hombre*", Augusto Roa Bastos.

CHAPTER

8

Conclusions

This thesis contributes to the development of distributed management systems based on intelligent agents and market mechanisms for DENs. To this end, both architectonic and algorithmic novel works have been performed and evaluated.

One of the key quests of this thesis is to facilitate, in a realistic manner, the adoption of software agents as a means to represent users in the virtual places that are emerging as result of the advances in information and communication technology. This is certainly a concern shared by the research community, which, aware of the lack of practical success of intelligent agents in environments similar to those posed by the Smart Grid and the computational grids, begins to wonder openly "*Where are all the intelligent agents?*" (see Section 3.1.2, page 51). This thesis argues that part of the reason for the failure to apply this technology lies in the recurring manner in which agent-based solutions are designed, essentially consisting of installing software agents in distributed nodes, delegating to each of them the responsibility for conducting the control, negotiation and coordination tasks that the problem or the business opportunity requires. That is, in the common solution, local agents are envisioned as highly capable entities from a technological and behavioral standpoint. This approach has been stretched to the limit in the case of the Smart Grid, which requires accessing external data services, as well as developing advanced negotiation and coordination tasks in an environment highly reactive and sensitive to failures.

One of the aims of this thesis is to find an architectural solution to this problem, so that software agents become a credible alternative for the management of the electrical networks. For this purpose, this research begins by studying similar well-trodden paths, using the **analogy** as a valuable research resource. The result of this part of the study is the novel concept of **Agency Services model**, which is completely based on application fields and technologies with very similar goals. In particular, this novel architectural model inherits many important features from the **cloud computing** paradigm, which has evolved from being a *buzzword* to becoming the type of solution demanded by the users. The solution is also enriched with architectonic and organizational foundations of other application fields, which nowadays face similar challenges, especially advances coming from P2P networks, virtual organizations and computational grids are present. The Agency Services model therefore can be seen as an architectonic innovation in the area of intelligent agents arising from the adaptation of seminal ideas and application of successful advances in analogous fields. Furthermore, by virtue of this procedure, it is ensured that the solution can be completely implemented using present-day technology, so it is not necessary to speculate about developments to come. Surprisingly, the use of analogies as an approach and opportunity had not been exploited in this field so far, so it is advisable that future research should take more advantage of this approach's potential, since when it comes to highly sensitive contexts such as the Smart Grid, all knowledge of successful practices in virtual distributed environments may prove critical. However, it should not be forgotten that, even though the Agency Services model has been drawn up with the Smart Grid in mind, it can be equally applied to other fields, including computational grids themselves.

In the specific case of the Smart Grid, the Agency Services model, through the ASPEM entity, draws a simpler but equally powerful architecture. It is principally intended to: (*i*) free customers from all demanding tasks typically associated with the individual management of end-nodes, such as accessing external data services and developing complex social behaviors; and (*ii*) empower customers to participate in advanced energy markets while holding simple infrastructures in households. In practice, these characteristics represent an important step on the road to a more participatory, accessible and less error-prone Smart Grid. Also, they must be seen as a logical step from a practical and economic standpoint. As for the functionality, it must be highlighted that the Agency Services model, thanks to the interaction between local and broker agents, is able to retain all features of agent-based solutions.

The **review of the literature** presented in Chapter 4 confirms that, despite the lack of practical success of this type of solution, most of the studies are essentially based on installing software agents in the consumption and production points. Somehow, this recurrent way of dealing with the problem demonstrates a lack of both practicality and self-criticism. Another interesting concluding remark related to the state of the art has to do with the **multidisciplinary** nature of the subject: with the need to combine solutions coming from both the artificial intelligence and electrical engineering fields. In this regard, most of the studies coming from the former research area seem to neglect important aspects, such as the adoption of standards and the characterization of energy markets. In particular, taking **standards** as a reference system is critical in order to avoid speculating on future scenarios, which diminishes the rigor of the proposal and gives rise to concepts that may not receive further support. In this regard, it is noteworthy the effort that Section 2.2 (Chapter 2) is dedicated to uniformizing and characterizing many concepts that the community dedicated to intelligent agents, with an evident lack of consensus, has developed in regard to this subject. After all, the research activity, if sincere, must be built and framed in a form that helps to unite efforts. However, the lack of precision does not exclusively belong to the teams in the artificial intelligence field, as similar faults can be detected in research coming from electrical engineering teams. Specifically, in this latter case research usually misapplies intelligent agents, so that erroneous behaviors and forms of interaction are assigned to them. Thus, it results that real multidisciplinary teams are not actually participating in a large part of the research efforts; instead, there are usually teams from one of the areas looking for new fields of application, which, as previously mentioned, has ended up affecting the success of agent-based solutions.

The review of the state of the art reveals that only the algorithms *CONSEC* and *mPJ* have been designed to handle the special characteristics of energy markets, among which are the need to continuously balance production and consumption, and the acceptance of combinatorial bids of both complementary and supplementary bids. However, as previously discussed, both algorithms are designed with opposite approaches: *CONSEC* looks for the equilibrium price by using a centralized scheme, whereas *mPj* sets forth a scenario in which each consumer holds its own auction, thus leading to a complete distributed solution based on software agents. The decision of which algorithm to use for the management of a particular DEN can be imposed by external factors: while *mPJ* facilitates including variables apart from energy prices, such as users' preferences, or the need for providing more decision-making capacity to users, the *CONSEC* algorithm provides a deterministic and controllable structure. From the point of view of this thesis, framed in the context of artificial intelligence and software agents, without doubt the proposal based on the *mPJ* algorithm is the most interesting and challenging. A revealing fact of this part of the investigation, which is certainly related to the difficulties that the multidisciplinary nature of the problem adds, is that, despite the prominence of *mPJ* over the rest of the agent-based methods, until the development of this thesis, this algorithm had not been evaluated using an electrical grid simulator, thus its real efficiency and qualities remain to be confirmed. The review of the state of the art also reveals the disappointing fact that doing reproducible research and studies based on standards is not a priority, both facets being essential for the research activity to become really constructive and serve as foundation of future work.

The multidisciplinary nature of the subject also concerns the **simulation infrastructure**, which must support two research fields that, due to their length and complexity, demand specialized tools. As described in Chapter 5, this challenge is addressed through a **co-simulation** solution, in which GridLAB-D and Jade, the most widely used tools in each area, are interconnected through a well-defined interface based on RESTful services. This approach, though it may involve a longer learning effort, has proven to be powerful and effective, reaching a high level of detail in both contexts, as well as a clearer and more modular design of the simulation infrastructure. It is also worth mentioning that the connection and synchronization of the two simulators, which is the most sensitive task in this matter, was enormously simplified by the fact that both are mature open source projects.

On the other hand, the **experiments** in Chapter 6, which have the mission of evaluating the properties of ASPEMs, prove that indeed this role is able to transfer many of the benefits of the cloud computing paradigm to the Smart Grid. Specifically, simulations show that end-nodes, regardless of the type of signal sent by the system operator, thanks to the mediation of broker agents, are able to successfully manage their devices and meet all their needs by using signals corresponding to the simplest profile of the OpenADR standard, which in practice results in an important saving of financial and human resources. Furthermore, the experiments demonstrate that ASPEMs open the door to the management of DR signals through advanced market mechanisms, thus enabling the active participation of customers in the management of energy resources, which is what Smart Grid claims to do. In particular, this study shows that, by virtue of the Agency Services model, customers can easily participate in parallel auction markets, which are known to be a particularly challenging solution for individual nodes, as they are required to manage complex processes. At this point it is necessary to emphasize the fact that markets can be held means a victory over an important restriction of the *Energy Interoperation* standard, which establishes that nodes of type VEN cannot communicate directly with each other. In addition, the fact that all these milestones are achieved in environments so restricted as the DR programs demonstrates the flexibility provided by the ASPEM role.

The experiments in Chapter 6 are also a valuable resource for gathering information about parallel auctions as management system of distributed contexts such as the Smart Grid and computational grids. Up until recently, the ability of this mechanism to scale and address the more advanced features of auctions had been known, but not its effectiveness in large distributed environments. In this regard, the present study is the first to warn about the enormous impact that a poor distribution of the bids can have on the effectiveness of the method as a management system. To this end, Chapter 7 develops the buyers' distribution mechanism **HUDP**. According to the experiments, it reaches levels of distribution that ensure the effectiveness of parallel auctions as a valid system for balancing energy. A key feature of HUDP is that its design is guided by rules intended to maintain markets competitive and preserve fundamental rights of users. Furthermore, it is tailored to meet the conditions of environments inhabited by a large number of nodes, with special emphasis on supporting concurrent access and providing short reaction time. Actually, in view of the high effectiveness of HUDP, it could be concluded that the poor distribution of buyers was the root cause of the bad results from the *mPJ* algorithm in Chapter 6, and therefore it can now be stated that providing a mechanism as set forth is as important as facilitating the implementation of parallel auctions. Upon its inclusion in the management system of the Smart Grid, it is expected that the system operator wants to control aspects of the distribution mechanism in order to strengthen the reliability and security of the network, although this may affect the competitiveness of the solution.

Finally, a fundamental premise of this thesis was to create work within the canons of *reproducible research*. In this regard, some important features to mention are: (*i*) the source code of the simulation infrastructure is open source and accessible from public repositories; (*ii*) it is documented, including detailed instructions to run simulations; and (*iii*) the research is evaluated in standard scenarios. The sole purpose of all these points is that the work done in this thesis can be contrasted; and, if it is of interest, can be extended by other researchers, thus fulfilling a final contribution: being part of something larger and from everyone.

8.1 Contributions

The research developed in this thesis makes the following contributions to the state of the art:

1. Chapter 2 questions the scheme traditionally proposed for the adoption of intelligent agents as a management system for DENs. This chapter sheds light on important technological and architectural challenges that this approach will have to face in realistic scenarios. Moreover, in order to learn how to overcome these obstacles, this chapter is the first to draw an analogy between DENs and other technological fields with architectural and behavioral similarities, such as P2P networks, virtual organizations and computational grids. In particular, the characteristics that have driven the most successful implementations in these research areas are highlighted, thus obtaining guidelines for the design of new architectural models that allow the reliable use of software agents in distributed energy environments. The contribution set forth in this chapter of the thesis was published in the paper "*Challenges of using smart local devices for the management of the Smart Grid*", in IEEE International Conference

on Smart Grid Communications, 2011 [LRHT11a]. This has been cited in [Sta12].

2. Chapter 3 presents the *Agency Services* model as a suitable architecture for facilitating the adoption of agent-based solutions in distributed, complex environments. This is the first research to propose intelligent agents as Cloud Computing services, as well as put forward the benefits this approach can bring to the software agents' community. This approach was published in the paper "*Software agents as Cloud Computing services*", in Advances on Practical Applications of Agents and Multiagent Systems, Springer, 2011 [LRHT11b]. This has been cited in [Tal12], [HDlPJ+12], [HdlPR+12], [ZK13], [Tal14] and [NEGR15]. Furthermore, a preliminary version of this work was published in the paper "*Agency Services: an agent-based and services-oriented model for building large virtual communities*" [ILR10]. This other paper has been cited in [Pal12].
3. Also in Chapter 3, the Agency Services model is adapted to the particular case of the Smart Grid. In particular, ASPEMs, a new type of intermediary node, is presented as a means to achieve the full integration of software agents in the management system of DENs. This section of the chapter also shows the benefits that using broker agents can bring to the implementation of distributed energy markets, which thus far were designed to be implemented through the direct interaction of end-nodes, ignoring all the difficulties that this approach entails. Parts of the ideas described in this regard were published in the paper "*Agent-based services for building markets in distributed energy environments*", in Renewable Energy & Power Quality Journal (ICREPQ), 2010 [LRHT10].
4. Chapter 4 presents a novel review of the most relevant algorithms that have been proposed for using distributed energy markets in DENs. Besides commenting on the operation and main characteristics of each work, these are studied and classified according to the features that future energy markets will require. The classification helps to identify important gaps in the research efforts, and gives a new perspective of the suitability of them for meeting the forthcoming challenges. As a result of this work, the article "*Methods for*

the management of distributed energy environments using software agents and market mechanisms: A survey" [LRHTar] was submitted to the journal Electric Power Systems Research. At this moment, the article has been accepted with revisions.

5. Chapter 6, apart from evaluating the benefits of incorporating supernodes in the Smart Grid architecture, to the best of our knowledge, is the first study to simulate parallel auction markets in the energy context, and therefore the first to give insight into both the actual efficiency and the drawbacks of this mechanism. In this sense, the study confirms that the efficacy presumed to parallel auctions may be canceled if bids are not uniformly distributed among sellers. Also, by implementing simulations entirely based on standards, the content of the chapter serves to confirm that software agents is a valid mechanism for the management of distributed energy environments. The work related to the evaluation of the ASPEM role in OpenADR programs was published in the article "*Infrastructure based on supernodes and software agents for the implementation of energy markets in demand-response programs*", in the journal Applied Energy, 2015 [LRHT15]. The information on this publication is provided in Table 8.1 and Table 8.2, including the impact factor and ranking of the corresponding journal.
6. Chapter 7 introduces the mechanism HUDP, which uniformly distributes the participation of buyers in large distributed environments based on parallel auctions. This is the first solution to deal with the lack of efficiency that, in terms of the management, produces the poor distribution of bids. The solution is tailored to advanced auction settings, supporting combinatorial bids, as well as bids expressed in form of linear piece-wise functions. This chapter offers a detailed study of the parameters and conditions that affect the behavior and effectiveness of any solution intended to solve this kind of problem. All of this study was published in the article "*Regulation of the buyers' distribution in management systems based on simultaneous auctions and intelligent agents*", in the journal Expert Systems with Applications, 2015 [LRHTHC15]. The information on this publication is provided in Table 8.3 and Table 8.4, including the impact factor and ranking of the corresponding journal.

Title	*Infrastructure based on supernodes and software agents for the implementation of energy markets in demand-response programs*
Authors	I. Lopez-Rodriguez, M. Hernandez-Tejera
Journal	Applied Energy
Impact	5.613 (2014)
Year	2015
Volume	158
Pages	1–11
DOI	http://dx.doi.org/10.1016/j.apenergy.2015.08.039

Table 8.1: Meta description of the article [LRHT15], developed as part of this dissertation, and published in the journal *Applied Energy*.

Area	Rank	Quartile
Energy & Fuels	2 / 88	Q1
Engineering Chemical	6 / 134	Q1

Table 8.2: Ranking of the journal *Applied Energy*.

8.2 Future work

Throughout this thesis several issues have arisen that, because of space and focus, have not been fully developed, but these nevertheless could lead to new and interesting contributions to the matter, and that therefore deserve more attention in the form of future works. This section outlines them.

- It is common that, when restoring from a DR event, the joint activity of all consumption devices gives rise to demand peaks. In order to avoid this effect, the OpenADR standard defines a window time over which devices choose at random when to restore their operation. Thus, it is expected that devices restoration is uniformly distributed along the window time. In this regard, ASPEMs,

Title	*Regulation of the buyers' distribution in management systems based on simultaneous auctions and intelligent agents*
Authors	I. Lopez-Rodriguez, M. Hernandez-Tejera, J. Hernandez-Cabrera
Journal	Expert Systems with Applications
Impact	2.240 (2014)
Year	2015
Volume	42
Pages	8014 – 8026
DOI	http://dx.doi.org/10.1016/j.eswa.2015.06.019

Table 8.3: Meta description of the article [LRHTHC15], developed as part of this dissertation, and published in the journal *Expert Systems with Applications*.

Area	Rank	Quartile
Computer Science, Artificial Intelligence	29 / 123	Q1
Engineering, Electrical and Electronic	48 / 249	Q1
Operations Research & Management Sciences	12 / 81	Q1

Table 8.4: Ranking of the journal *Expert Systems with Applications*.

by taking advantage of the global vision they have, are in position to develop more intelligent mechanisms: using the information provided by the broker agents, ASPEMs can intelligently decide the best time for a node to return to its normal operating mode. Therefore, it is advisable to study the contribution ASPEMs can make in this matter, as well as to check if this alternative approach yields better results than that proposed by the standard.

- As discussed in Section 6.2, as a result of the interaction model imposed by the OpenADR standard, experiments conducted in Chapter 6 cannot use a common market for all ASPEMs. Instead, each ASPEM is required to hold its own market in which only its own broker agents participate. In principle, this circumstance causes loss of opportunities and competitiveness. Therefore, it is still necessary to conduct new simulations that evaluate how this fact is really affecting the efficiency of both the management system and the market. Also, it is necessary to study the minimum changes needed to apply to the OpenADR standard, in order that it supports the creation of global markets in which broker agents of all ASPEMs can participate jointly.

- With regard to the management system based on priorities (see Section 6.1), whenever a node is selected by the algorithm, before asking others, all the available negative load of that selected node is used to try to meet the gap of demand. This approach, therefore, tends to build solutions composed of few nodes with consumption levels quite restricted. It would be interesting to find out if solutions based on a greater number of nodes with higher consumption levels could yield better results for the security and reliability of the network. In this case, it would be also interesting to design solutions capable of reaching a compromise between the grid's reliability and user rights.
- As for auction markets, the work of this thesis is mainly concerned with their ability of them to achieve reliable and efficient management systems, thus pushing all matters relating to energy pricing into the background. Accordingly, it is still necessary to devote more efforts to this issue, including how price levels should be defined in order to achieve a control scheme that ensures minimum targets.
- The HUDP mechanism does not address the particularities of combinatorial bids. Therefore, it is still necessary to evaluate the impact that this type of bids has on the efficiency of the result. It is necessary to: (*i*) conduct an extensive experimental evaluation of the effect of such bids in the efficiency of the HUDP solution; and, if necessary, (*ii*) introduce changes that improve the outcome of the solution for these cases.
- Simulations are run in the scenario *IEEE 13-node*, which contains 629 households. In the future it would be interesting to perform simulations with thousands of households in order to learn more about the management systems implemented in Chapter 6. Given that parallel auctions is a solution ready to scale, it would be particularly interesting to study whether there are significant variations in the efficiency of the HUDP mechanism. This type of experiment

will require the installation of the simulation infrastructure in systems for high performance computing, such as supercomputers.

Appendices

APPENDIX

A

XML schemas

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.siani.es/agencyservices/aspem
    /1.0" elementFormDefault="qualified" xmlns="http://www.w3.org
    /2001/XMLSchema" xmlns:tns="http://www.siani.es/agencyservices
    /aspem/1.0">
    <element name="aspem">
      <complexType>
        <sequence>
          <element name="code" type="string"></element>
          <element name="name" type="string"></element>
          <element name="host" type="string"></element>
          <element name="port" type="int"></element>
        </sequence>
      </complexType>
    </element>
</schema>
```

Listing A.1: XML schema definition of an ASPEM.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.siani.es/agencyservices/
    userPreferences/1.0"
```

```
elementFormDefault="qualified" xmlns="http://www.w3.org/2001/
    XMLSchema" xmlns:tns="http://www.siani.es/agencyservices/
    userPreferences/1.0">

<element name="userPrefs">
  <complexType>
    <sequence>
      <element name="levelsSchedule" type="tns:userPrefsType"
          minOccurs="0" />
      <element name="priority" type="tns:priorityType" minOccurs=
          "0" />
      <element name="easyLevel" type="tns:levelType" minOccurs="0
          " />
      <element name="hardLevel" type="tns:levelType" minOccurs="0
          " />
      <element name="loadMode" type="tns:loadModeType" minOccurs=
          "0" default="none" />
      <element name="levelsPrices" type="tns:levelsPricesType"
          minOccurs="0" />
      <element name="startingPrice" type="float" default="0"
          minOccurs="0" />
    </sequence>
  </complexType>
</element>

<complexType name="userPrefsType">
  <simpleContent>
    <extension base="string">
      <attribute name="name" type="string" use="required"></
          attribute>
    </extension>
  </simpleContent>
</complexType>

<complexType name="levelsPricesType">
  <sequence>
    <element name="levelPrice1" type="tns:levelPrice" />
    <element name="levelPrice2" type="tns:levelPrice" />
    <element name="levelPrice3" type="tns:levelPrice" />
  </sequence>
</complexType>
```

```
  <simpleType name="priorityType">
    <restriction base="int">
      <minInclusive value="0" />
    </restriction>
  </simpleType>

  <simpleType name="levelType">
    <restriction base="int">
      <minExclusive value="-1"></minExclusive>
      <maxInclusive value="3"></maxInclusive>
    </restriction>
  </simpleType>

  <simpleType name="loadModeType">
    <restriction base="string">
      <enumeration value="none"></enumeration>
      <enumeration value="easy"></enumeration>
      <enumeration value="hard"></enumeration>
    </restriction>
  </simpleType>

  <simpleType name="levelPrice">
    <restriction base="int">
      <minInclusive value="0"></minInclusive>
    </restriction>
  </simpleType>
</schema>
```

Listing A.2: XML schema definition of user preferences.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.siani.es/agencyservices/
    simulationResults/1.0" elementFormDefault="qualified" xmlns="
    http://www.w3.org/2001/XMLSchema" xmlns:tns="http://www.siani.
    es/agencyservices/simulationResults/1.0">

  <element name="simulationResults">
    <complexType>
      <sequence maxOccurs="unbounded" minOccurs="0">
        <element name="market" type="tns:marketType"></element>
```

```
    </sequence>
    <attribute name="scenarioCode" type="string"></attribute>
    <attribute name="nMarkets" type="int"></attribute>
    <attribute name="usingStartingPrice" type="boolean"></
       attribute>
    <attribute name="usingRandomDistribution" type="boolean"></
       attribute>
    <attribute name="cteCF" type="boolean"></attribute>
  </complexType>
</element>

<complexType name="marketType">
  <sequence>
    <element name="brokerAction" type="tns:brokerActionType"
       maxOccurs="unbounded" minOccurs="1"></element>
  </sequence>
  <attribute name="start" type="date"></attribute>
</complexType>

<complexType name="brokerActionType">
  <choice>
    <element name="producer" type="tns:producerType"></element>
    <element name="consumer" type="tns:consumerType"></element>
    <element name="normal" type="tns:normalType"></element>
  </choice>
  <attribute name="id" type="string"></attribute>
  <attribute name="level" type="float"></attribute>
</complexType>

<complexType name="producerType">
  <sequence maxOccurs="unbounded" minOccurs="1">
    <element name="genBlock">
      <complexType>
        <attribute name="amount" type="float"></attribute>
        <attribute name="price" type="float"></attribute>
        <attribute name="to" type="string"></attribute>
      </complexType>
    </element>
  </sequence>
  <attribute name="placed" type="float"></attribute>
  <attribute name="capacity" type="float"></attribute>
```

```
  </complexType>

  <complexType name="consumerType">
    <sequence maxOccurs="unbounded" minOccurs="1">
      <element name="loadBlock">
        <complexType>
          <attribute name="amount" type="float"></attribute>
          <attribute name="price" type="float"></attribute>
          <attribute name="from" type="string"></attribute>
        </complexType>
      </element>
    </sequence>
    <attribute name="covered" type="float"></attribute>
    <attribute name="demanded" type="float"></attribute>
  </complexType>

  <complexType name="normalType"></complexType>
</schema>
```

Listing A.3: XML schema definition of simulation results

APPENDIX

B

Algorithms of experimental evaluation based on priorities

Require: *toCover* is the amount of load to be covered
Require: *brokers* is the list of available broker agents
  sort in descending order *brokers*
  covered $\leftarrow 0$
  assignments $\leftarrow$ empty list
  while *covered* $<$ *toCover* **do**
    broker $\leftarrow$ next broker of the list *brokers*
    if *broker* is in mode_easy **then**
      add (*broker* in easy_level) to *assignments*
      covered $\leftarrow$ *covered* + amount covered by *broker*
    end if
  end while
  return *assignments*

Algorithm 6: The demand can be covered using easy-loads.

```
Require: toCover is the amount of load to be covered
Require: brokers is the list of available broker agents
  sort in ascending order brokers
  for all broker in brokers do
    if broker is in (mode_normal or mode_easy) then
      add (broker in critical_level) to assignments
      covered ← covered + amount covered by broker
    else
      maxLevel ← maximum level allowed by broker
      add (broker in maxLevel) to assignments
      covered ← covered + amount covered by broker
    end if
  end for
  restart iterator of the list brokers
  while covered < toCover do
    broker ← next broker of the list brokers
    if broker is in mode_hard then
      add (broker in critical_level) to assignments
      covered ← covered + amount covered by broker
    end if
  end while
  return assignments
```

Algorithm 7: The demand can be covered without using hard-loads.

Require: *toCover* is the amount of load to be covered
Require: *brokers* is the list of available broker agents
targetLevel ← level required to cover *toCover*
// Firstly, all the easy loads are assigned
sort in descending order *brokers*
while *covered* < *toCover* **do**
broker ← next broker of the list *brokers*
if *broker* is in mode_easy **then**
maxLevel ← maximum level allowed by *broker*
if *maxLevel* > *targetLevel* **then**
maxLevel ← *targetLevel*
end if
add (*broker* in *maxLevel*) to *assignments*
covered ← *covered* + amount covered by *broker*
end if
end while
// Normal loads are assigned
sort in ascending order *brokers*
while *covered* < *toCover* **do**
broker ← next broker of the list *brokers*
if *broker* is in mode_normal **then**
add (*broker* in *targetLevel*) to *assignments*
covered ← *covered* + amount covered by *broker*
else if *broker* is in mode_hard **then**
maxLevel ← maximum level allowed by *broker*
if *maxLevel* > *targetLevel* **then**
maxLevel ← *targetLevel*
end if
add (*broker* in *maxLevel*) to *assignments*
covered ← *covered* + amount covered by *broker*
end if
end while
return assignments

Algorithm 8: The demand has to be covered using hard loads.

No hay que buscar la palabra más justa, ni la palabra más bella, ni la más rara. Busca solamente tu propia palabra.
"*Dichos de Luder*", Julio Ramón Ribeyro.

APPENDIX

C

Resumen en español

C.1 Objetivo

El objetivo de esta tesis, titulada "*Servicios de Agencia para la gestión de redes de energía distribuida usando mercados de subastas paralelas*", es la elaboración de una solución arquitectónica y algorítmica para hacer posible la gestión de entornos de energía distribuida usando agentes software y mecanismos de mercado. La tesis se propone realizar contribuciones realistas y prácticas, evitando ubicar estas en escenarios que, por la incertidumbre que encierra el contexto, pudieren no materializarse. Es por esto que una de las premisas esenciales del trabajo aquí desarrollado es simular y verificar todas las propuestas en entornos realistas y estándares. Asimismo, el trabajo se desarrolla con el propósito de ser reproducible, de forma que este pueda ser contrastado y continuado por otros grupos de investigación.

C.2 Planteamiento y metodología

El sistema de energía eléctrica (en adelante también *red eléctrica*) evidencia claros signos de obsolescencia. La infraestructura actual, la cual apenas ha experimentado innovaciones en las últimas décadas, se reconoce incapaz de afrontar los importantes retos que le aguardan, entre los que destacan el fuerte incremento de la demanda, la falta de control sobre los precios, el agotamiento de los combustibles fósiles y el efecto de los gases invernadero. Además, el alto coste de los recursos necesarios para cubrir situaciones eventuales, así como la baja eficiencia de los sistemas de generación y transmisión, son factores que también llaman a la incorporación de nuevas soluciones estructurales y tecnológicas. En concreto, es necesario actuar sobre los siguientes objetivos estratégicos:

- *Sostenibilidad*: limitar el consumo de energía e impulsar el desarrollo de energías limpias que contribuyan a frenar el cambio climático.
- *Competitividad*: crear mercados competitivos y transparentes que ayuden a combatir la volatilidad de los precios, y que sean capaces de transmitir a los usuarios las necesidades reales del sistema.
- *Fiabilidad en el suministro*: asegurar el suministro mediante la implantación de mecanismos que ayuden a reducir tanto la demanda como la dependencia de fuentes de energía externas.

La comunidad investigadora coincide en que no existe una solución única capaz de hacer frente a todos los retos. Al contrario, se entiende que esta sólo puede surgir de la aplicación de múltiples iniciativas. Así es que las acciones destinadas a educar a la ciudadanía en el consumo responsable son tan importantes como encontrar nuevas fuentes de energía. En concreto, en este último apartado, una de las líneas de acción más prometedoras es la instalación de *Fuentes de Energía Renovable* (FER). Las FER, además de encarnar una reserva ilimitada de recursos limpios y sostenibles, también son identificadas como recursos valiosos para paliar la dependencia de fuentes externas de energía y, por tanto, como una vía para recuperar el control sobre los precios. Sin embargo, son fuentes de naturaleza estocástica, de forma que su disponibilidad y producción no es predecible. Esta condición las convierte en recursos de difícil gestión para los sistemas de control actuales, implicando un riesgo importante para la fiabilidad y seguridad de la red eléctrica. Actualmente, cuanto mayor es la presencia de las FER, mayor es el riesgo de que la red se colapse. Es por ello que, para aprovechar todo su potencial, es necesario crear sistemas de gestión modernos capaces de adaptarse a la naturaleza dinámica y distribuida de las nuevas fuentes de generación, incluyendo las *Fuentes de Energía Distribuida*

(FED), las cuales, gracias a los avances tecnológicos, han mejorado notablemente sus prestaciones y disponibilidad.

Como parte de los planes de futuro de la red eléctrica también se estima esencial incrementar la participación de los usuarios. En este sentido, una de las metas más inmediatas es extender los programas de *Demanda-Respuesta* (**DR**) a los hogares, sobrepasando así los ámbitos industrial y comercial. El objetivo es que los usuarios, en respuesta a órdenes del operador del sistema y a señales basadas en el precio de la energía, sean capaces de modular y desplazar su demanda. La participación de los usuarios se hace posible a través de contadores inteligentes (en adelante **AMI**[1]).

La contribución de los usuarios a la gestión de la red, incluyendo la posible inyección de energía por parte de unidades locales, se espera que dé lugar a una red eléctrica que, además de reactiva, distribuida e inteligente, también sea bidireccional. Este proyecto es comúnmente conocido como **Smart Grid**.

La enorme dependencia que las sociedades modernas tienen del consumo eléctrico sugiere emprender una transición gradual y segura hacia el Smart Grid. Como parte de este proceso de transformación se trabaja especialmente en el desarrollo de áreas de la red de distribución que, dotadas de sistemas de control modernos, sean capaces de gestionar eficazmente la presencia de unidades de generación distribuida. Estas áreas, que son conocidas genéricamente como ***redes de energía distribuida***, representan una oportunidad para integrar en un entorno acotado y escalable las tecnologías y sistemas de información sobre los que se afirmará el futuro Smart Grid.

En particular, en este campo de trabajo destaca el concepto de *microrred*, la cual se define como una agregación de cargas y unidades de generación distribuidas que operan como una sola unidad capaz de consumir y producir energía (Figura C.1). Desde el punto de vista del operador del sistema, las microrredes se comportan como un punto más del sistema, conformando así una vía eficaz para la integración gradual y transparente de las unidades de generación distribuida en la red eléctrica. Desde el punto de vista de la red principal, las microrredes son clasificadas como *buenos ciudadanos*, lo cual significa que no suman complejidad ni nuevos factores de riesgo al sistema de gestión. Asimismo, para fases más avanzadas, se espera que las microrredes se comporten como *ciudadanos modelo*, entendiéndose por ello la

[1]En inglés, *Advanced Metering Infrastructure*

capacidad de proporcionar servicios auxiliares a la red, como puede ser inyectar energía cuando sea necesario, o bien reducir la demanda.

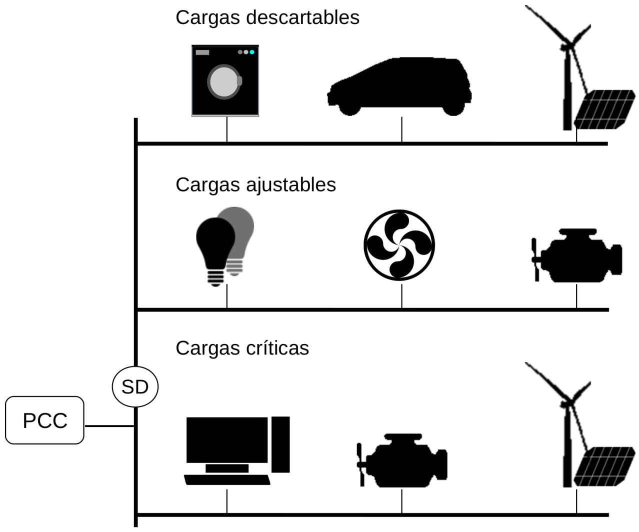

Figura C.1: Arquitectura básica de una microrred.

En el marco del proyecto europeo CRISP se ha desarrollado un concepto similar al de microrred, aunque bajo el nombre de *celda de energía*. En concreto, una celda de energía se define como un área acotada y autogestionada de la red de distribución diseñada para albergar fuentes de generación distribuida. Una de las características más representativas de las celdas de energía es que estas pueden ser agrupadas, de forma que la unión de dos o más de ellas da origen a una nueva celda. Esta propiedad configura una estructura capaz de escalar horizontal y verticalmente (Figura C.2). La principal diferencia, por tanto, entre los conceptos de microrred y celda de energía es que las primeras están pensadas para funcionar como una red compuesta de unidades simples, mientras que las segundas pueden contener otras celdas.

Como parte de su operación diaria, los entornos de energía distribuida deben satisfacer requisitos funcionales, económicos, medioambientales y legislativos. Es por ello que, aparte de los controles eléctricos de acción rápida, estos entornos requieren sistemas de control inteligentes. Estos se denominan EMS [1] y su objetivo

[1]En inglés, *Energy Management System*.

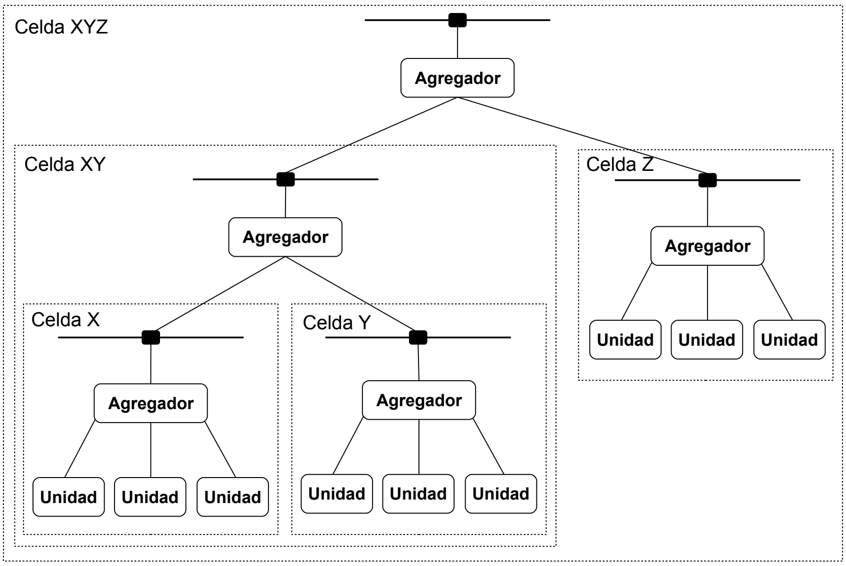

Figura C.2: Arquitectura basada en el concepto *celda de energía* del proyecto Europeo CRISP.

principal es optimizar la operación de la celda o la microrred a través de la planificación, coordinación y supervisión de los recursos. El EMS se enmarca en el área de control secundaria y efectúa la planificación de la actividad en base a factores tales como: condiciones procedentes de la red principal, características particulares de cada una de las unidades de generación, cantidad de demanda que puede ser desplazada o descartada, cantidad de energía que puede ser almacenada, el precio de la energía, la legislación vigente, estimaciones de la demanda y predicciones meteorológicas. Se debe notar que el EMS, más que a una unidad física, hace referencia a un concepto que, en la práctica, puede ser implementado desde la forma más sencilla, como son los mecanismos de acción manual, hasta la forma más sofisticada y compleja, como pueden ser los sistemas distribuidos basados en conceptos y técnicas propios de la inteligencia artificial.

En la práctica, el EMS se implementa mayoritariamente como un módulo centralizado basado en algoritmos de optimización. Sin embargo, este enfoque no resulta eficaz en entornos de tamaño medio y grande porque: (*i*) el coste computacional para encontrar una solución aumenta exponencialmente con el tamaño del modelo; (*ii*) no es capaz de tratar variables de origen estocástico y no lineal; (*iii*) es poco reactivo; y (*iv*) no proporciona autonomía ni poder de decisión a los usuarios, los cuales se ven limitados a expresar sus intenciones a través de funciones de precio o utilidad.

Al contrario, los sistemas de control distribuidos encajan mejor con la definición de Smart Grid, el cual pretende crear una red de energía bidireccional, distribuida, inteligente y reactiva. Precisamente con este objetivo, en el proyecto CRISP se desarrolla el modelo de gestión **Supply Demand Matching** (SDM), según el cual los propietarios de las unidades de generación y consumo negocian e intercambian dinámicamente bloques de energía sin comprometer la estabilidad de la red. A diferencia de los mecanismos clásicos de gestión de la demanda, el modelo SDM proporciona autonomía a las unidades de producción, las cuales pasan a tener capacidad de actuación. En esencia, el modelo SDM propone la gestión del entorno a través de micromercados de energía: los productores y consumidores intercambian bloques de energía que, en términos netos, logran mantener balanceada la red. Estos micromercados se diseñan para ser reactivos, instanciados bajo demanda y de corta duración (habitualmente inferior a 15 minutos).

La implementación del modelo de gestión SDM requiere la instalación en cada nodo de componentes software capaces de desarrollar comportamientos autónomos. Estas tienen la misión de: (*i*) representar los intereses y preferencias de los usuarios en los micromercados de energía; y (*ii*) cooperar y coordinarse con el resto de nodos para satisfacer las metas colectivas impuestas por el operador del sistema. Los agentes inteligentes, por sus características, son aceptados como la tecnología más apropiada para afrontar este reto, de forma que la implementación del EMS se propone habitualmente en forma de sistema multiagente.

Sin embargo, los agentes inteligentes no han sabido consolidarse como solución práctica en muchos de los entornos que, al igual que el Smart Grid ahora, requieren sus cualidades para implantar soluciones avanzadas. La realidad de esta tecnología es que, en contra de lo proyectado, y pese a que los nuevos sistemas de información configuran un entorno ideal, su aplicación sigue conservando un estatus minoritario. Este hecho causa que la brecha entre la actividad investigadora y su implementación en soluciones prácticas sea cada vez mayor. Tanto es así que, con el fin de generar debate sobre esta cuestión, voces importantes en la materia empiezan a preguntarse abiertamente ***¿Dónde están todos los agentes inteligentes?*** Aunque se puede replicar que, en muchos casos, los agentes inteligentes se hallan enmascarados en procesos internos, también es cierto que el impacto de las soluciones basadas en los mismos se halla lejos de lo esperado. En particular, destaca la ausencia de agentes inteligentes que representen a los usuarios en muchos de los entornos virtuales que han emergido a raíz de las nuevas tecnologías de la información, como son los sitios de subastas o los sitios de compra y venta en internet. Asimismo, los agentes inteligentes no han logrado asentarse como solución de referencia en entornos a priori ideales, como son las redes P2P, las *redes computacionales*[1], las organizaciones virtuales o la web semántica.

Parte del escaso éxito de los agentes inteligentes pasa por la complejidad que implica articular soluciones basadas en la coordinación y cooperación de entidades autónomas en entornos compartidos y cohabitados. Conscientes de esta barrera, se han destinado numerosos esfuerzos a la creación de utilidades software que facilitan la programación de soluciones basadas en agentes. Sin embargo, este enfoque requiere que los usuarios desarrollen sus propios agentes, distanciándose así del tipo de solución que, en la práctica, predomina en el mundo del software, el cual se conduce cada vez más a soluciones que liberan a los usuarios de todas aquellas tareas que no comprendan la simple utilización de la aplicación. Por tanto, para que los agentes inteligentes sean entendidos como una tecnología capaz de ofrecer soluciones realistas y prácticas, antes de ser aplicados en el Smart Grid incidiendo en errores del pasado, es necesario que la comunidad investigadora diseñe nuevos modelos que faciliten su adopción.

Para lograr la implementación de la gestión de entornos de energía distribuida usando el modelo SDM, el presente trabajo de investigación emplea la siguiente **metodología**:

- Estudia el estado del arte dedicado a las soluciones basadas en agentes inteligentes para la gestión de entornos de energía distribuida. A partir de este estudio, extrae las barreras tecnológicas y arquitectónicas que aún afrontan las soluciones actuales.

[1]También conocido como *Sistema de computación distribuido.*

- Estudia áreas tecnológicas que enfrentan retos muy parecidos, como pueden ser las redes computacionales, las redes P2P y las organizaciones virtuales. A continuación, a partir de las soluciones más exitosas en estas áreas, extrae las lecciones más importantes para el diseño de nuevos modelos arquitectónicos que faciliten la integración realista de los agentes inteligentes en el sistema de gestión de las redes de energía distribuida.
- Analiza las características del paradigma *Cloud Computing*, el cual definitivamente se ha asentado como un modelo exitoso de implantación y distribución de software. Además, como parte del trabajo desarrollado en este apartado, se extraen las propiedades y lecciones más importantes de este nuevo modelo.
- Partiendo de las conclusiones obtenidas en los análisis anteriores, diseña un nuevo modelo arquitectónico que facilita la adopción de los agentes inteligentes en entornos tecnológicos distribuidos. Asimismo, se estudia la aplicación de este nuevo modelo al caso concreto de los sistemas de gestión del Smart Grid.
- Estudia los algoritmos más importantes de la literatura que, basados en mecanismos de mercado, se proponen para la gestión de redes de energía distribuida. Analiza el funcionamiento de cada propuesta y las clasifica conforme a las características que demandan los mercados de energía. A continuación, identifica las propuestas más prometedoras y las facetas en las que aún es necesario invertir nuevos esfuerzos.
- Construye una infraestructura de *simulación conjunta* mediante la coordinación de herramientas de simulación propias de las redes eléctricas y los sistemas multiagente. El objetivo es lograr una herramienta de simulación que haga posible la evaluación de las contribuciones presentadas en este trabajo de forma realista y estándar.
- Evalúa la aplicación del nuevo modelo arquitectónico usando la infraestructura de *simulación conjunta* elaborada en el apartado anterior, y el algoritmo que, conforme al análisis del estado del arte, resulta más completo y apropiado. La evaluación se realiza en el entorno de los programas DR, ya que estos suponen uno de los hitos más inmediatos del Smart Grid.

- Realiza los ajustes y mejoras necesarias en el plano algorítmico para lograr una solución con la fiabilidad y la eficiencia que requieren las redes de energía distribuida.

Con este proceso se busca y se logra realizar contribuciones prácticas en el área de la gestión de redes de energía distribuida, tanto en el plano arquitectónico como en el algorítmico.

C.3 Aportaciones originales

C.3.1 Retos de los sistemas de gestión de energía basados en agentes inteligentes

El diseño de los sistemas de gestión distribuidos se basa en la instalación de controladores inteligentes en los nodos locales. A estos se les asignan tareas como (Figura C.3): la participación en mercados de energía, la interacción con el usuario, la monitorización de los recursos de generación y consumo, y el acceso a servicios externos de datos. Aunque desde un punto de vista teórico los agentes inteligentes son capaces de desempeñar todas esas funciones, en la práctica la adopción de soluciones basados en ellos oculta riesgos tecnológicos y estructurales que aún no han sido estudiados en detalle, siendo este trabajo de investigación el primero en hacerlo. En particular, los **riesgos estructurales** son:

- En contra de lo proyectado, la instalación de agentes software en dispositivos similares a los contadores inteligentes no configura un sistema de control flexible y reactivo. Debido a la complejidad del sistema eléctrico, y en particular a la fiabilidad que este exige, en un escenario así sería necesario actualizar y depurar los agentes software periódicamente. En el caso de redes de tamaño medio y grande, esta tarea, además, se proyecta como costosa y lenta.
- Las interfaces de usuario que ofrecen este tipo de unidades para la monitorización y configuración de los recursos son pobres y limitadas. Además, este tipo de solución complica y encarece su fabricación y mantenimiento. En la práctica, para que estos dispositivos puedan ofrecer una interfaz web moderna sería necesario completar los mismos con conectores de red y servidores web

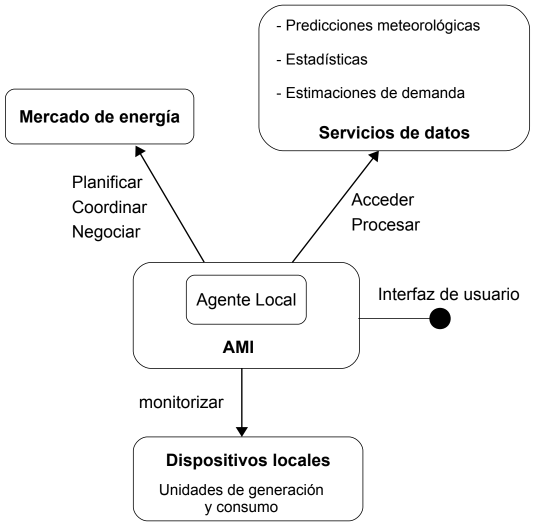

Figura C.3: Tareas e interacciones que se asocian a los dispositivos de control locales en el marco del sistema de gestión del Smart Grid.

embebidos; acciones que, por otra parte, incidirían negativamente en el precio y mantenimiento del producto final.

- Según el estándar *Energy Interoperation* (EI) de OASIS, los nodos sólo pueden comunicarse directamente con otros nodos que hagan las veces de *agregador*, o bien que retransmitan las señales del operador del sistema. Esta norma impide crear sistemas multiagente completamente distribuidos, obligando a adoptar soluciones que, en esencia, son centralizadas.

Por otra parte, los **riesgos tecnológicos** son:

- Para que los agentes puedan planificar la actividad de las unidades de producción y consumo es necesario que estos accedan a fuentes externas de datos tales como predicciones meteorológicas, estimaciones de la demanda y precios de las fuentes de energía. Acceder a esta clase de servicios y procesar adecuadamente su información son actividades demasiado exigentes para un dispositivo como el AMI.
- Aunque los agentes inteligentes son una tecnología que se propone habitualmente para la implementación de sistemas de gestión distribuidos, lo cierto es que en la práctica triunfan soluciones que, pese a ser menos capaces, resultan

más efectivas y menos complejas. En consecuencia, es necesario estudiar esta falta de éxito y desarrollar nuevos modelos que faciliten la adopción de los agentes inteligentes como solución.

C.3.2 Lecciones aprendidas de entornos tecnológicos similares

Las **redes P2P** comparten con el Smart Grid estructura y objetivos similares. En particular, el objetivo de las redes P2P es facilitar el intercambio de recursos entre nodos que, en el apartado funcional, pueden ser considerados como iguales o *pares*. Un factor que ha probado ser decisivo en el éxito de la redes P2P es la topología, la cual, a grandes rasgos, se puede clasificar como (Figura C.4):

a) *Centralizada*: los pares se conectan a nodos centrales que desempeñan funciones complejas, como la indexación y búsqueda de contenidos.

b) *Descentralizada*: todos los nodos son responsables de desempeñar todas las funciones.

c) *Híbrida*: se basa en nodos especiales, denominados *supernodos*, los cuales, para una subsección de la red, funcionan como punto de entrada y desempeñan funciones avanzadas, incluyendo la búsqueda global de contenidos.

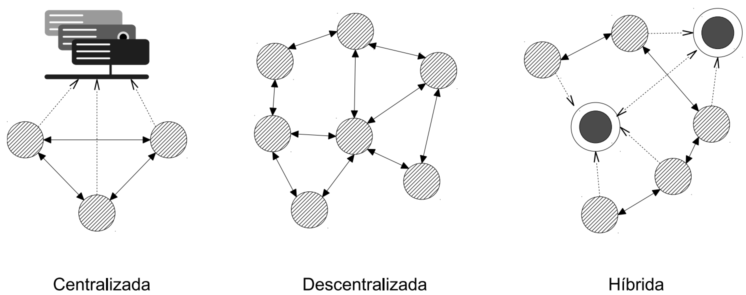

Figura C.4: Tipos principales de topología de las redes P2P.

La desventaja más notable de la topología centralizada es que consta de un punto de fallo que hace que la red sea demasiado vulnerable a ataques o averías. Aunque la topología descentralizada soluciona este problema, en este caso, la ausencia de un índice global de todos los contenidos ocasiona que las búsquedas sean bastante más ineficaces. En el caso concreto de *Gnutella*, la cual es la red P2P descentralizada más conocida, los mensajes de búsqueda se envían recursivamente a los nodos vecinos hasta alcanzar un nivel de profundidad predefinido. En la práctica, este proceso ha demostrado saturar la red de mensajes y dar origen a resultados sesgados. Para mitigar estos efectos, en Gnutella han emergido de forma natural nodos capaces de indexar una gran cantidad de contenidos y conexiones. Estos nodos, en realidad, desempeñan el mismo rol que los supernodos en las topologías híbridas, refrendando así esta última propuesta. De hecho, la topología híbrida es la más adoptada, disponiendo de muchas implementaciones de éxito que demuestran su eficiencia para el intercambio de recursos. La razón de su éxito reside en no asumir que todos los nodos son iguales: lo cierto es que, en la práctica, los nodos disponen de diferentes propiedades y, por tanto, resulta natural que haya nodos que asuman más responsabilidades que otros.

Las funciones y retos de las **redes computacionales** se asemejan bastante a los del Smart Grid. En concreto, su propósito es crear una infraestructura distribuida de súper computación a través de la interconexión de recursos heterogéneos. La primera generación de redes computacionales se caracterizó por aportar soluciones ad-hoc difíciles de exportar. Más adelante, en la conocida como *segunda generación*, surgieron numerosos marcos de desarrollo que tenían por objetivo facilitar la implementación de esta clase de sistemas. Sin embargo, las soluciones creadas a partir de dichos entornos mostraron ser monolíticas, poco escalables y difíciles de integrar con otros sistemas, dando lugar a redes aisladas e incapaces de interaccionar entre sí. No obstante, cabe destacar que las redes computacionales de estas dos generaciones, pese a los defectos que se les atribuyen, lograron ser funcionales y efectivas, cumpliendo con el cometido con el que fueron diseñadas.

La tercera y, hasta el momento, última generación de sistemas de computación distribuida nace con un enfoque fuertemente orientado a servicios, distinguiéndose especialmente por reemplazar el concepto de *recurso* por el de *servicio*. Esta generación, además, destaca por intensificar los esfuerzos en la creación y adopción de estándares, así mejorando la interoperabilidad, y haciendo posible la implementación de sistemas de control distribuidos. En particular, estos últimos sistemas, al igual que en el Smart Grid, tienen como objetivo proporcionar un rol más prominente a los usuarios, los cuales, a través de agentes inteligentes, pueden negociar dinámicamente la prestación de los recursos en base a factores tales como la demanda o la calidad del servicio.

Una **Organización Virtual** (OV) es una coalición de entidades distribuidas que colaboran y comparten recursos con el fin de cumplir metas tanto colectivas como particulares. En general, el concepto de OV se usa como enfoque para abordar la problemática de entornos que se caracterizan por ser cambiantes, ágiles y distribuidos, en los que las partes buscan alianzas para aumentar sus competencias y reducir riesgos. En concreto, las sociedades que habitualmente se describen como OV comparten las siguientes propiedades: (*i*) son creadas con el fin particular de aprovechar una oportunidad de negocio temporal; (*ii*) hacen uso intensivo de las nuevas tecnologías de la información; (*iii*) las partes que integran la organización se encuentran distribuidas; (*iv*) usan mecanismos de cooperación y coordinación; y (*v*) los socios son autónomos, de forma que, además de las metas globales, buscan el beneficio particular. Conforme a la definición de OV, incluyendo las características citadas, las redes de energía distribuida basadas en mecanismos de mercado pueden considerarse OV, siendo esta tesis el primer trabajo en señalarlo.

La experiencia ganada en este campo advierte de que la automatización del ciclo de vida de una OV es una tarea compleja que, en la práctica, requiere tanto de soluciones específicas como de la participación de operadores humanos. En particular, la mayor parte de la complejidad se concentra en la etapa de *creación* de la OV, que se compone de tareas como: la identificación de las oportunidades de negocio, el diseño del plan de acción, la selección de los socios que participarán en el proceso de negocio y la asignación de tareas a los mismos. Además, algunos entornos como el Smart Grid añaden retos típicos de contextos abiertos y reactivos. Entre estos se encuentra la necesidad de comunicar y coordinar eficazmente agentes heterogéneos, y la implementación de mecanismos que ayuden a evaluar la capacidad y efectividad de los socios. Como solución a todos estos retos, la comunidad dedicada al estudio y desarrollo de OV propone la creación de entornos especiales, denominados *Virtual Breeding Environment* (**VBE**), cuyo fin es simplificar la instanciación dinámica de OV. En concreto, los VBE se pueden definir como un grupo estable de socios bien conocidos dispuestos a asociarse para explotar oportunidades de negocio eventuales. Los VBE constan de una autoridad responsable de certificar las capacidades de los socios que forman parte del grupo, garantizando así que pueden desempeñar eficazmente las tareas que se les encomienden. Es necesario hacer notar que no todos los socios del VBE siempre forman parte de todas las VO que se pudieren instanciar; en cada caso, según la oportunidad de negocio y las características de los socios, se elige el grupo de ellos que mejor se adecúe a las metas a lograr. Otros componentes importantes de los entornos VBE son la instalación de una infraestructura tecnológica común, y la definición de ontologías, protocolos de comunicación y convenciones sociales. Todos estos elementos ayudan a crear un entorno fiable y normalizado que facilita las actividades propias de la etapa de creación.

En resumen, las **lecciones** más importantes que cabe extraer de los esfuerzos invertidos en las tres áreas anteriores son:

i. Estudiar si la instalación de súper nodos en los entornos de energía distribuida puede aportar ventajas similares a las reflejadas en las redes P2P, tales como la simplificación de las tareas que tradicionalmente se asocian al resto de nodos. Es preciso apuntar que este tipo de esquema es compatible con los estándares de OASIS para el Smart Grid.
ii. Diseñar soluciones orientadas a servicios y basadas en estándares con el fin de garantizar la interoperabilidad y escalabilidad del sistema de gestión.
iii. Alcanzar un compromiso entre la prestación de funciones avanzadas y el pragmatismo de las mismas.
iv. Establecer marcos VBE para facilitar la interacción segura de los agentes software, incluyendo los procesos de negociación y coordinación entre ellos.

C.3.3 Servicios de Agencia

Concepto

Esta tesis desarrolla el modelo **Servicios de Agencia**, el cual tiene como objetivo facilitar, de forma realista y práctica, la integración de los agentes inteligentes en los nuevos entornos virtuales.

Partiendo de la idea central del paradigma Cloud Computing, el modelo **Servicios de Agencia** (SA) propone que empresas externas con el conocimiento y los recursos tecnológicos necesarios ofrezcan agentes software que, después de ser contratados como servicios, participen en entornos virtuales representando a los usuarios. En el modelo, estos agentes software se denominan **agentes bróker**, y se diseñan para desarrollar planes de acción y conducir diálogos que tienen por objeto la gestión del entorno y/o negociar el intercambio de recursos. Para los dispositivos e instalaciones del cliente, el modelo SA propone la instalación de agentes ligeros capaces de comunicarse con el agente bróker. Estos se denominan **agentes locales** y son responsables de tareas sencillas, tales como la aplicación de comandos consignados por el agente bróker, enviar información a este último sobre el estado de los recursos locales, o bien informarle sobre nuevas directivas del usuario.

En esencia, el modelo SA propone distribuir las funciones que habitualmente se atribuyen a un sólo agente, de forma que la representación del usuario es acometida por un agente remoto en continua comunicación con uno o más agentes locales. Es necesario destacar que, para no perder la simplicidad que caracteriza al modelo Cloud Computing, tanto los agentes locales como el agente bróker son suministrados por el proveedor de servicios localizado en la nube. A este respecto, es importante que la instalación del agente local sea automatizada, de forma que no requiera la intervención del usuario.

Conforme a lo expuesto, el modelo SA se compone de las siguientes figuras (Figura C.5):

- *Proveedor de Servicios de Agencia (PSA)*: una empresa que posee el conocimiento y la infraestructura necesaria para desplegar agentes software que representen a los usuarios en entornos virtuales.
- *Sitio de Negocio*: entidad externa que instancia los entornos virtuales en los que participan los agentes bróker de los PSA. Los PSA autorizados a desplegar o interactuar con el sitio de negocio son previamente certificados por una autoridad. Los agentes bróker pueden ser desplegados en el sitio de negocio, o bien interactuar con este a través de interfaces bien definidas. A modo de ejemplo, entornos virtuales susceptibles de aplicar y aventajarse de este modelo son los sitios de compra y venta en internet, los sistemas de computación distribuida, las redes P2P y el Smart Grid.
- *Cliente*: entidad que contrata los servicios del PSA con el objeto de participar en los entornos virtuales instanciados en los sitios de negocio. A resultas de

su participación, el cliente espera completar tareas específicas u obtener algún tipo de beneficio.

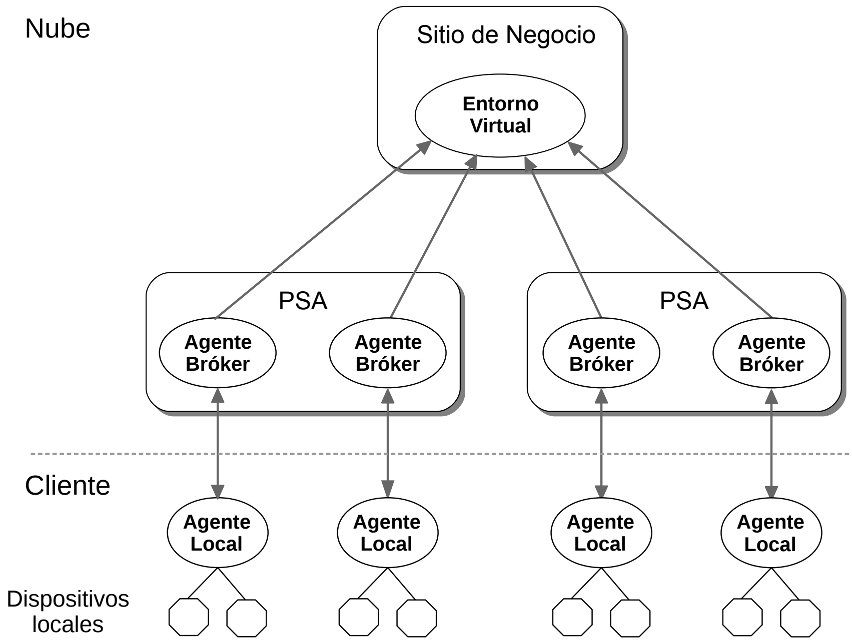

Figura C.5: Esquema general del modelo de Servicios de Agencia.

Para ilustrar la utilidad del modelo SA, a continuación se describe un ejemplo basado en el sitio de subastas y compra y venta *eBay*. En concreto, eBay, con el fin de aumentar la competitividad y automatización de los procesos, podría definir interfaces a través de los cuales agentes software puedan realizar ofertas y recibir información acerca de las subastas en activo. En este caso, PSA autorizados por eBay podrían ser contratados por los usuarios para automatizar su participación en las subastas, eliminando así la necesidad de estar pendientes de las ofertas que les interesan o de desarrollar sus propias soluciones. Los PSA, a su vez, significarían una garantía para eBay, ya que estos asegurarían que los agentes cumplen con convenciones sociales y reglas de negocio específicas. En este ejemplo, eBay es el *sitio de negocio*, los *entornos virtuales* son las subastas que este celebra, y los usuarios son los clientes, los cuales, en este caso, no disponen de recursos locales. Otros entornos que están en disposición de beneficiarse del modelo SA, como pueden ser el

Smart Grid y los sistemas de computación distribuida, sí cuentan con recursos locales y, por tanto, requieren de la intervención de agentes locales que los monitoricen y que apliquen las órdenes enviadas por los agentes bróker.

Un cliente puede tener asociado más de un agente local. Por ejemplo, es posible usar un agente local para informar al agente bróker del estado de los recursos gestionados, y asimismo usar agentes locales instalados en dispositivos móviles para recibir información del bróker, o bien enviarle para nuevas directivas. El agente bróker, así, además de participar en las sociedades virtuales, también puede hacer las veces de *agente proxy*, ya que funciona de nexo entre los agentes locales.

El usuario puede configurar sus preferencias a través de interfaces web o aplicaciones de dispositivos móviles, ambas proporcionadas por el PSA. Esta información es transmitida al bróker correspondiente una vez instanciado. De este modo, en entornos con recursos locales, como puede ser el Smart Grid, no sería necesaria la instalación de interfaces embebidas.

Una vez el cliente haya contratado los servicios de un PSA, las fases principales por las que atraviesa la participación de este son:

- *Iniciación*: el sitio de negocio informa a los PSA de que, en respuesta a una oportunidad de negocio, una nueva sociedad virtual ha sido instanciada. Cada PSA despliega los agentes bróker que, en representación de los clientes, participarán en la misma. Si es necesario, el PSA actualiza de forma automatizada y dinámica el software de los agentes locales. A continuación, el agente bróker le comunica a los agentes locales que un nuevo proceso de negocio ha comenzado. Si es necesario, el agente local informa al agente bróker sobre el estado de los recursos locales.
- *Ejecución*: el agente bróker se registra en el entorno virtual y, de acuerdo con el estado de los recursos locales y las directrices marcadas por el usuario, negocia con otros agentes bróker desarrollando, si es necesario, planes de acción. A lo largo de este proceso el agente bróker puede informar periódicamente a los agentes locales sobre el proceso de participación en el entorno virtual. Además, cuando existan recursos locales, como es el caso del Smart Grid y los sistemas de computación distribuida, el agente bróker transmite comandos para su gestión. Por su parte, el agente local informa al agente bróker sobre eventos locales y nuevas directivas definidas por el usuario.

- *Cierre*: el agente bróker registra toda la información relacionada con su participación en la sociedad virtual, e indica a los agentes locales que el proceso ha finalizado.

También es posible una versión más sencilla del modelo de Servicios de Agencia. En concreto, el sitio de negocio y el PSA pueden compartir el mismo nodo de la infraestructura, de forma que tanto los entornos virtuales como los agentes bróker son provistos por la misma entidad (Figura C.6). Aunque es cierto que con esta versión del modelo se pierde competitividad, la solución aún conserva características atractivas, ya que los PSA siguen ofreciendo a los usuarios la opción de configurar cómo se deben comportar los agentes bróker.

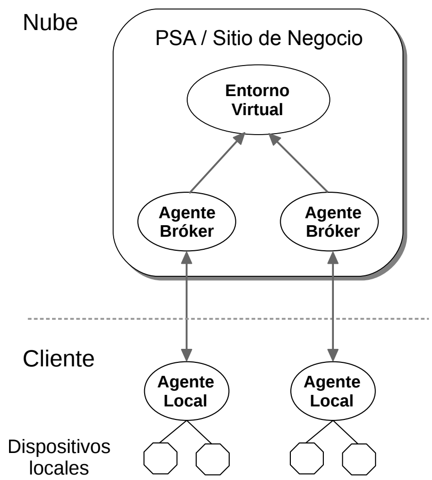

Figura C.6: Esquema general de la versión simplificada del modelo de Servicios de Agencia.

Al ofrecer a los agentes inteligentes como un servicio más de la nube, tal y como propone el modelo de Servicios de Agencia, se obtienen los siguientes **beneficios**:

i. La complejidad que implica desarrollar agentes software capaces de participar en entornos virtuales se delega en empresas externas. Asimismo, la necesidad de actualizar los agentes para mejorar su rendimiento, o bien adaptarlos a una nueva interfaz, es responsabilidad de los proveedores de servicios.

ii. Los usuarios pueden contratar e interactuar con los agentes usando cualquier dispositivo con conexión a internet, y por tanto pueden participar en sociedades virtuales sin perder movilidad.

iii. Los usuarios pagan por funcionalidades específicas, de forma que son ellos quienes determinan el ámbito y las habilidades que los agentes bróker pueden desarrollar en las sociedades virtuales.

El modelo SA replica muchas de las características que han conducido a otras tecnologías a convertirse en soluciones de éxito. En particular:

- Los PSA son entornos VBE. Como se ha descrito, los sitios de negocio evalúan y certifican la capacidad de los PSA que pretenden participar en las sociedades virtuales. Esta condición garantiza que todos los agentes software desplegados por los PSA cumplen las normas de comportamiento y que, asimismo, usan ontologías y tecnologías de comunicación comunes.
- El modelo SA presenta una estructura muy similar a la de las redes P2P híbridas, las cuales se caracterizan por incluir el concepto de supernodo. De hecho, se puede afirmar que, en la práctica, los PSA son supernodos que proporcionan funciones avanzadas de *brokering* a otros nodos. Asimismo, pueden ofrecer servicios de procesamiento y acceso a datos.
- Tal y como sugiere la experiencia obtenida en el área de los sistemas de computación distribuida, el nuevo modelo tiene una fuerte orientación a servicios y logra soluciones que, pese a estar basadas en tecnologías avanzadas, no dejan de ser prácticas. Esto último es posible gracias a que el modelo de SA delega las tareas más complejas a empresas especializadas del sector, y a que además logra configurar un entorno acotado y seguro.

Comparado con las soluciones que tradicionalmente se aplican en el mundo de los sistemas multiagente, el modelo de Servicios de Agencia mejora la participación, la escalabilidad, la fiabilidad y la competitividad del entorno.

Servicios de Agencia para el Smart Grid

Para la gestión inteligente y reactiva del Smart Grid, este trabajo propone la instalación de entidades denominadas *Proveedores de Servicios de Agencia para la Gestión de Energía* (**ASPEM**[1]). En concreto, los ASPEM son nodos inteligentes que ofrecen, en forma de servicio, agentes bróker que los usuarios contratan para participar activamente en el sistema de gestión de entornos de energía distribuida

[1]En inglés, *Agency Servivces Provider for the Energy Management*

(especialmente en aquellos basados en negociaciones y mecanismos de mercado). Los ASPEM no son simples *agregadores*: los mecanismos de control de estos últimos se basan en la redirección de las señales de entrada, o bien en la aplicación de las mismas conforme a reglas predefinidas; mientras que un nodo ASPEM, a través de los agentes bróker, hace posible la implementación de algoritmos distribuidos basados en negociaciones y acciones de coordinación y cooperación.

Uno de los objetivos principales de los Servicios de Agencia es simplificar la carga de trabajo asociada al nodo cliente y, en consecuencia, simplificar la infraestructura tecnológica que este necesita. Este trabajo demuestra que dicho principio puede aplicarse con éxito en el campo de las redes de energía distribuida. En particular, los ASPEM y la acción de los agentes bróker facilitan que los nodos cliente puedan beneficiarse de las funciones de los perfiles más avanzados del estándar OpenADR mediante dispositivos propios del perfil más simple. En la práctica, por ejemplo, esta cualidad facilita que señales de tipo *delta*, después de ser procesadas por los agentes bróker, puedan ser convertidas en señales de tipo *simple*, que son las que finalmente reciben los hogares después de pasar por los ASPEM.

Los servicios de datos son de especial importancia para la gestión del Smart Grid. Entre ellos destacan aquellos que proporcionan estimaciones de la demanda, precios de la energía y predicciones meteorológicas. En los esquemas tradicionales de gestión se asume que el acceso a estos servicios, así como el procesamiento de su información, debe ser asumido por los AMI de los clientes. Sin embargo, estas son tareas demasiado exigentes para dispositivos que pretenden ser instalados de forma masiva y que, por tanto, se espera que sean asequibles y fáciles de mantener. Esta clase de trabajo es más propia de centros de datos, en cuya categoría, de hecho, se inscriben los ASPEM. Esta información, después de ser generada o bien consumida por terceros, se transmite a los agentes bróker para que puedan desarrollar una gestión adecuada de los recursos que representan.

La Figura C.7 describe las interacciones de un nodo ASPEM según la descripción proporcionada en esta sección. En los escenarios entrevistos para el Smart Grid, un nodo ASPEM podría representar miles, cientos de miles o incluso cifras más altas de nodos finales. Los ASPEM ofrecerían servicios de *brokering* y datos, ayudando a simplificar la infraestructura de los nodos cliente, y capacitándolos para defender sus intereses mediante comportamientos inteligentes.

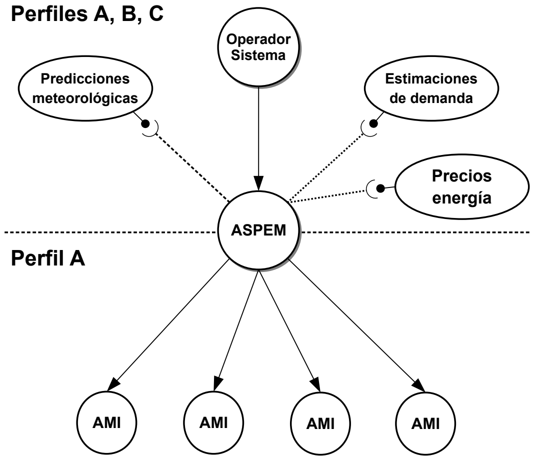

Figura C.7: Interacciones de un nodo ASPEM.

C.3.4 Mercados de energía en programas DR

La implantación de los programas DR es uno de los objetivos más inmediatos del Smart Grid. Esta tesis, con el fin de aportar soluciones prácticas, centra los experimentos en ellos.

En general, en los programas DR los participantes se limitan a aplicar señales para la restricción del consumo, no disponiendo de otra capacidad de acción o réplica. En consecuencia, para poder instalar sistemas de gestión basados en mecanismos de mercado, y así explotar todo el potencial que puede brindar el modelo de Servicios de Agencia, primero es necesario introducir los conceptos de *consumidor* y *productor*. Esta tarea es posible porque, de hecho, en los programas DR algunos roles o comportamientos pueden interpretarse como propios de consumidores y productores. Estos son:

- *Carga negativa*: un nodo que intencionadamente consume menos de lo esperado o consignado. Bajo este comportamiento, el nodo puede considerarse un productor porque genera un déficit de demanda que puede ser consumido por otros nodos.
- *Carga crítica*: un nodo que, incluso bajo circunstancias especiales, requiere o desea que se le garantice una cantidad mínima de suministro. En entornos como los programas DR, en los que se asume que todos los nodos deben li-

mitar su consumo al unísono, las cargas críticas pueden interpretarse como consumidores debido al exceso de demanda que generan.

Sobre la base de estos conceptos, con el fin de habilitar la creación de mercados en los programas DR, en este trabajo se introducen los siguientes conceptos:

- *Easy-load*: nodo dispuesto a descartar una cantidad determinada de energía durante un período específico de tiempo.
- *Hard-load*: nodo que necesita proteger una cantidad determinada de consumo durante un período específico de tiempo.
- *Normal-load*: nodo que no aplica ningún comportamiento especial y, por tanto, dispuesto a aplicar la señal DR de entrada.

En un entorno donde estos roles son posibles, las cargas de tipo *hard-load*, con el fin de no aplicar las señales DR ordenadas por el operador, luchan por adquirir los bloques de energía que ofrecen las cargas de tipo *easy-load*. En el caso concreto de los programas OpenADR, las cargas *hard-load* evitan asumir el nivel de consumo consignado por una señal de tipo *simple* (*moderate*, *high* y *critical*). En este punto es importante hacer notar que una carga sólo está en disposición de no aplicar un nivel si logra adquirir energía suficiente para cubrir toda la demanda extra que dicha acción genera.

C.3.5 Estado del arte de los EMS basados en agentes inteligentes y mecanismos de mercado

En la tesis, con el fin de analizar apropiadamente la eficacia de los algoritmos de mercados destinados a la gestión de energía distribuida, antes de adentrarse en el estudio del estado del arte, se pasa a caracterizar los mercados de energía. Este análisis preliminar demuestra ser necesario al observar que gran parte de las propuestas fallan al evaluar la verdadera naturaleza y complejidad de los mercados de energía. En concreto, estos se deben clasificar como mercados multiunidad y multielemento que requieren de la presencia de ofertas combinadas de tipo complementario y suplementario.

En este punto es preciso aclarar que los mercados de energía se componen de varias unidades de tiempo, conformando cada una de ellas un elemento de negociación. En concreto, los mercados se definen como multielemento porque es habitual que los consumidores y productores estén interesados en realizar ofertas que abarcan varias unidades de tiempo. Asimismo requieren aceptar ofertas combinadas porque la evaluación de los elementos que componen la oferta debe realizarse en relación al conjunto.

La presencia de pujas combinadas, pese a la complejidad que añaden a la resolución del problema, es una característica de especial importancia para los mercados de energía. Por un lado, algunos tipos de unidades de generación, como las basadas en combustibles fósiles, es necesario que operen durante un período mínimo de tiempo para resultar rentables. Asimismo, algunos tipos de unidades de consumo, como la lavadora o el lavavajillas, habitualmente necesitan de varias unidades de tiempo para poder completar su carga de trabajo. En estos casos, si las pujas de elementos complementarios no fuesen soportadas, los usuarios estarían obligados a enviar por separado ofertas para cada unidad de tiempo implicada en el proceso, acción que supone un riesgo importante para el usuario, ya que este necesita que todas las unidades sean aceptadas o rechazadas conjuntamente. Por otra parte, las pujas de elementos suplementarios facilitan que dispositivos como las lavadoras o los termos eléctricos puedan definir varios períodos de tiempo en los que su carga puede ser suministrada, de forma que el operador del sistema decida, de entre las opciones disponibles, cuál es la más conveniente para la red. Esta clase de pujas, en general, permite suavizar los períodos de demanda máxima. Sin embargo, pese a la importancia que, como ilustran los ejemplos, tienen las pujas combinadas en los mercados de energía, estas habitualmente son ignoradas en el estado del arte.

A continuación se resumen las características de los métodos más empleados en la literatura, así como los algoritmos más completos y prometedores.

- *Subastas dobles*: los productores y consumidores envían sus ofertas a una autoridad central que determina, en base a ellas, cómo se han de distribuir los recursos. Aunque a priori esta metodología es capaz de asumir todas las propiedades de los mercados de energía, en la práctica es difícil obtener algoritmos que resuelvan el mercado de forma eficiente y que asimismo sean capaces de escalar satisfactoriamente. De hecho, en entornos grandes esta aproximación tiende a resultar en problemas de tipo NP-hard. Como resultado, es habitual que los autores opten por mecanismos de resolución simples, como la localización del precio de equilibrio. El estudio del estado del arte revela que esta aproximación se usa esencialmente en pequeñas microrredes, y que se obvian funcionalidades importantes, como la capacidad de negociar múltiples elementos al mismo tiempo.

- *Subastas paralelas*: cada vendedor ejecuta su propia subasta, de forma que varias de ellas pueden operar al mismo tiempo. Como las subastas habitualmente no son *dobles*, sino simples, la complejidad para hallar soluciones eficaces es menor, incluso cuando estas soportan la definición de ofertas combinadas. En el estado del arte sobresale el algoritmo **mPJ**, el cual, además de soportar todas las propiedades de los mercados de energía, demuestra ser escalable.
- *Búsqueda orientada a precios*: los participantes expresan sus preferencias a través de funciones de precios, siendo este, el precio, el factor que determina qué cantidad de energía ha de consumir y producir cada nodo. En la práctica, el precio que resuelve de forma óptima el mercado no puede hallarse con métodos analíticos, lo que conduce a ejecutar procesos de búsqueda muy exigentes desde el punto de vista computacional. Como resultado, los trabajos optan por soluciones sencillas, como es el cálculo del precio de equilibrio. Al igual que las *subastas dobles*, los trabajos que adoptan este enfoque omiten características esenciales de los mercados de energía, como son la capacidad de negociar múltiples elementos al mismo tiempo, o incluso la posibilidad de especificar múltiples unidades.
- *Búsqueda orientada a recursos*: la búsqueda de la solución se realiza en el espacio de recursos. En este caso, se dice que el sistema está en equilibrio cuando se halla una distribución de los recursos óptima de Pareto. Comparada con la búsqueda orientada a precios, la ventaja de esta aproximación es que la solución se puede hallar analíticamente a partir de las funciones de utilidad. Este mecanismo ha recibido bastante atención por parte de la comunidad investigadora, destacándose entre ellos el proyecto CRISP de la Unión Europea. Entre los mecanismos propuestos, en especial destaca el algoritmo **CONSEC**, capaz de soportar pujas combinadas. *CONSEC* ha sido simulado en entornos realistas con buenos resultados, siendo uno de los mecanismos más prometedores y completos para la implementación de mercados de energía.

- *Asignación simétrica*: algoritmo que, basándose en la valoración que guardan los participantes de cada uno de los recursos, realiza una asignación iterativa buscando siempre el máximo beneficio. Aunque esta aproximación es utilizada en numerosas ocasiones en el estado del arte, consta de limitaciones que hace difícil su aplicación en mercados de energía de tamaño medio o grande. Además de que no es posible tratar más de un elemento al mismo tiempo, las simulaciones demuestran que el algoritmo no escala bien, siendo necesario limitar el número de iteraciones y, con ello, la calidad y potencia de la solución.

La Tabla C.1 presenta una clasificación de los trabajos más destacados del estado del arte. Esta ayuda a diferenciar las características de los mercados de energía que satisface cada propuesta, así como la forma en las que han sido validadas.

Tabla C.1: Descripción de los algortímos más destacados del estado del arte para la gestión de entornos de energía distribuida usando mecanismos de mercado.

Work	Multi unidad	Multi elem.	Comb. Comp.	Comb. Supl.	Test	Algoritmo
Dimeas et al. (2005)	sí	no	no	no	sim.	Asignación simétrica
Funabashi et al. (2008)	sí	no	no	no	sim.	Asignación simétrica
Nunna and Doolla (2013)	sí	no	no	no	sim.	Asignación simétrica
Dimeas et al. (2004)	sí	—	—	—	test	Subasta doble
Ramachandran et al. (2011)	sí	—	—	—	sim.	Subasta doble
Arnheiter (2000)	sí	no	no	no	test	Equilibrio
Logenthiran (2008)	sí	no	no	no	test	Equilibrio
Ygge and Akkermans (1996)	sí	no	no	no	test	Equilibrio
Ygge and Akkermans (2000)	sí	sí	no	no	sim	Equilibrio
Carlsson and Andersson (2007)	sí	sí	sí	sí	sim	Equilibrio
Amin and Ballard (2000)	sí	sí	—	—	test	Subastas paralelas
Penya and Jennings (2005)	sí	sí	sí	sí	test	Subastas paralelas
Rahman et al. (2007)	sí	no	—	—	sim.	Matchmaker

A partir de esta clasificación se puede concluir que los únicos algoritmos capaces de cubrir todas las necesidades de los mercados de energía son *mPJ* y *CONSEC*.

Ambas propuestas se basan en aproximaciones opuestas: *CONSEC* usa una estrategia centralizada que busca la asignación óptima de Pareto de los recursos; mientras que *mPJ* propone que cada consumidor celebre su propia subasta, componiendo así una solución distribuida basada en agentes autónomos. A *CONSEC*, pese a su buen rendimiento en las simulaciones, se le pueden atribuir las desventajas típicas de los algoritmos centralizados de búsqueda, las cuales principalmente son:

i. Los usuarios sólo pueden expresar sus condiciones a través de funciones de utilidad.
ii. El algoritmo asume que existe una entidad externa con capacidad infinita de generación y consumo.
iii. Los comportamientos de los agentes sólo pueden configurarse en base a señales de precios.
iv. En caso de que un agente cambie su plan de acción, o bien varíen las condiciones del contexto, el algoritmo de búsqueda debe volver a ejecutarse, afectando así a los planes de acción de todos los nodos.

Por su parte, *mPJ* aún debe ser simulado en el entorno de las redes de energía y comprobar, con ello, la eficiencia del mismo. En particular, es necesario evaluar cómo afectan los inconvenientes que normalmente se atribuyen a las subastas paralelas, los cuales principalmente son: (*i*) mala distribución de los compradores entre las subastas disponibles; (*ii*) la incapacidad de los compradores para usar estrategias basadas en el *overbooking* de su potencia de generación; y (*iii*) necesidad de que agentes locales sean capaces de gestionar el ciclo de vida completo de una subasta. En este último aspecto, el modelo de Servicios de Agencia puede significar una gran ayuda, puesto que la responsabilidad de celebrar y conducir las subastas recae sobre los agentes bróker.

También cabe reseñar que gran parte de los trabajos del estado del arte no han sido simulados en entornos realistas, y que ninguna solución está basada en los estándares propuestos para implementar modelos de interacción en el Smart Grid.

C.3.6 Infraestructura de simulación

Para la simulación del sistema de energía eléctrica se utiliza GridLAB-D; un potente simulador desarrollado por el *Departamento de Energía de EE.UU.* (DOE) para afrontar los próximos retos en las áreas de generación, transmisión, distribución y consumo de energía. GridLAB-D está desarrollado bajo los términos del software libre y dispone de una fuerte comunidad de usuarios. GridLAB-D ofrece la oportunidad de ejecutar simulaciones sobre escenarios estándares. En concreto, los experimentos de este trabajo se ubican en el bus *IEEE-13*, el cual se compone de 629 hogares que cuentan con termos eléctricos, unidades de aire acondicionado y sistemas de iluminación. Para dar cabida a los conceptos introducidos en este documento, incluyendo los relacionados con los mercados de energía, la sintaxis de GridLAB-D para modelar escenarios fue extendida con un módulo nuevo denominado *AgencyServices*.

Los ASPEM son implementados como aplicaciones Java que se ejecutan en servidores Jetty, los cuales se inician automáticamente con cada simulación. Por su parte, la plataforma de agentes bróker se implementa usando *Jade*, el cual es el entorno de desarrollo más utilizado por la comunidad dedicada a la implementación y simulación de sistemas multiagente. En particular, Jade es conocido por facilitar la implementación de soluciones basadas en el estándar FIPA. En la infraestructura de simulación presentada en este documento cada ASPEM posee un contenedor FIPA donde se despliegan y ejecutan los agentes bróker. Asimismo, todos los diálogos entre agentes se implementan usando interacciones estándares. En concreto, los agentes usan mensajes de tipo *FIPA Request* para solicitar la ejecución de acciones; mensajes de tipo *FIPA Query* para solicitar información específica; y mensajes de tipo *FIPA Inform* para informar sobre estados o hechos del contexto. Además, todos los conceptos, predicados y acciones involucrados en las interacciones del sistema se formalizan a través de ontologías.

El módulo *AgencyServices* añade al vocabulario de GridLAB-D el elemento "`ASBox`", término que funciona como abreviatura de *Agency Services Box*. Este elemento, que está pensado para formar parte del AMI, representa el entorno en el que se ejecuta el agente local y, por tanto, representa el punto de conexión entre el nodo y los ASPEM. Los elementos ASBox reciben mensajes enviados por los agentes bróker. En particular, estos envían mensajes del tipo "*OadrDistributeEvent*" del estándar OpenADR, cuyo propósito es especificar el nivel de consumo que debe adoptar el hogar. Como se ha descrito, los hogares sólo reciben señales del perfil más simple, cuyos valores se implementan de la siguiente manera: *normal*, no se requiere que el hogar restrinja su consumo; *moderate*, las unidades de aire acondicionado deben desconectarse; *high* las unidades de aire acondicionado y los termos deben desconectarse; *critical*, todos los dispositivos de consumo, incluidas las luces, deben desconectarse. En realidad, esta definición de las señales es demasiado agresiva, de forma que, en escenarios reales, es común proponer implementaciones que varían el nivel de confort. No obstante, en este trabajo se opta por una definición como la expuesta porque permite centrar la atención en los mecanismos de gestión, y porque asimismo proporciona curvas de demanda en las que es más fácil distinguir el efecto de las señales aplicadas.

Los agentes bróker se comunican con el agente local a través del protocolo XMPP, siendo este uno de los mecanismos propuestos en el estándar OpenADR. Por tanto, como resultado, la infraestructura de simulación incluye un servidor XMPP. Otro componente a destacar de la infraestructura es el repositorio de estimaciones de demanda. Este es necesario para que los nodos dispongan de indicadores que les informen de la cantidad aproximada de energía que consumirán durante un período específico de tiempo y para cada uno de los niveles de consumo posibles.

La Figura C.8 muestra los componentes principales de la infraestructura de simulación, incluyendo cada uno de los componentes software que se usan para implementarlos

C.3.7 Simulación de mercados de subastas paralelas usando nodos ASPEM

Como se puede concluir del análisis del estado del arte, el mecanismo que mejor se adapta a la naturaleza distribuida del Smart Grid, y que asimismo cumple todos los requisitos de los mercados de energía, son las subastas paralelas inversas. Bajo este esquema, un usuario que desea proteger su demanda es un usuario que ofrece bloques de demanda en el mercado (subastador o vendedor); mientras que un usuario dispuesto a ofrecer parte de su demanda es un usuario dispuesto a pujar para cubrir la demanda de terceros (comprador). De esta forma, usando los conceptos definidos en la Sección C.3.4, las cargas de tipo *hard* son típicas de los subastadores, y las de tipo *easy* de los postores. En este sentido, se debe notar que el rol desempeñado por el agente puede cambiar entre mercado y mercado, cuya duración en

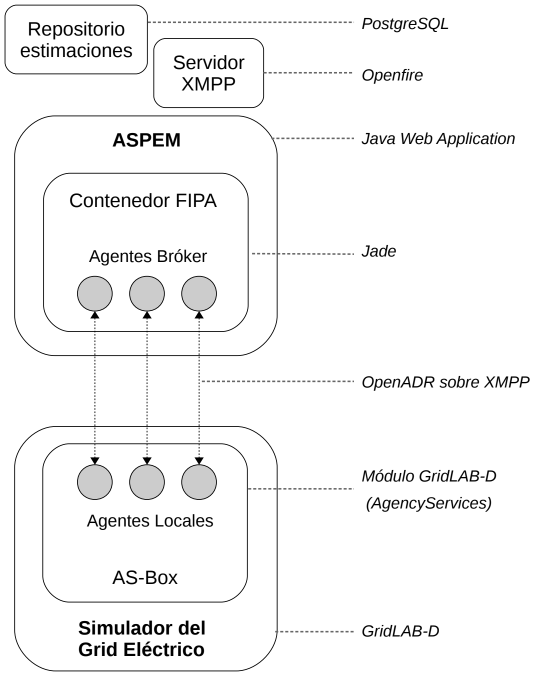

Figura C.8: Componentes de la infraestructura de simulación.

las simulaciones se establece en 30 minutos. Por consiguiente, los eventos con una duración superior a 30 minutos se gestionan a través de secuencias de mercados.

Después de recibir un evento OpenADR, los ASPEM instancian los agentes bróker que representan a los clientes en la gestión del evento. A continuación, cada bróker registra en un directorio FIPA los roles que desempeñará en cada uno de los mercados que componen el evento, los cuales pueden ser: vendedor (subastador), comprador o ninguno (si decide no participar). Posteriormente, los consumidores (vendedores) consultan el directorio para obtener la lista de productores (compradores) disponibles. Los consumidores que no encuentran suficientes productores para cubrir un nivel completo de la señal OpenADR (*moderate*, *high* o *critical*) deben cancelar la subasta y aplicar la señal OpenADR de entrada.

Después de recibir las invitaciones, los productores deciden en qué subastas participar. Es importante aclarar que los productores no pueden participar en todas las subastas simultáneamente porque esto seguramente implicaría pujar por encima de sus posibilidades reales. Es decir, supondría llevar a cabo una estrategia basada en el concepto de *overbooking*, la cual supone un riesgo para el comprador y, en general, para el sistema. Por último, los consumidores deciden qué ofertas son aceptadas. Como resultado, los productores a los que se hayan aceptado ofertas deberán adoptar niveles de consumo más restrictivos, mientras que los consumidores que hayan logrado cerrar acuerdos podrán aumentar o mantener su nivel de consumo frente a la señal OpenADR de entrada.

Las simulaciones se realizan en un día típico de verano (1 de agosto de 2000) usando el escenario estándar IEEE-13. En este se incluyen dos ASPEM que prácticamente gestionan el mismo número y tipo de nodos (Tabla C.2), los cuales se configuran para actuar como cargas de tipo *normal*, *easy* o *hard* durante el período completo que abarca cada mercado. En concreto, las cargas *hard* se configuran para proteger toda su demanda (intentan conservar su consumo habitual); mientras que las cargas *easy* se definen para estar dispuestas a descartar, si es necesario, toda su demanda.

	normal (kW)	**moderate (kW)**	**high (kW)**	**critical (kW)**	**Easy load (kW)**	**Hard load (kW)**
aspem 1	61.137	27.103	15.552	0	4.670	32.652
aspem 2	66.461	30.871	15.194	0	5.819	34.891

Tabla C.2: Consumo de los ASPEM por cada nivel de consumo propio de las señales de tipo *simple* de OpenADR.

La Tabla C.3 describe el perfil del mercado. Como indica su contenido, cuando el valor de la señal es *moderate*, la relación entre la capacidad de puja y la cantidad subastada (en adelante R_{ca}) es $2{,}29$. Por consiguiente, cuando los subastadores no establecen precio de entrada, se puede afirmar que la oferta duplica la demanda. La tabla también muestra que, en este caso, la relación entre el número de productores y consumidores (en adelante R_{np}) es $3{,}10$. Cuando el valor de la señal es *hard*, la situación es menos ideal, ya que R_{ca} es igual a $0{,}91$. Es decir, la cantidad subastada es superior a la capacidad de compra; o, lo que es lo mismo, la oferta es mayor que la demanda. Se debe notar que todos estos valores son menores cuando existen precios de entrada, puesto que la oferta de un comprador puede no ser válida para todas las subastas.

Señal	Demanda (kW)	Oferta (kW)	R_{ca}	R_{np}
moderate	15271	35114	2,29	3,10
hard	20416	18569	0,91	2,27

Tabla C.3: Perfil del mercado de subastas.

El mercado de subastas pasa por las siguientes etapas:

i. *Anuncio*: los ASPEM informan a los agentes bróker de la instanciación de un nuevo ciclo de mercados para gestionar eventos DR. Específicamente, se informa acerca de la duración del evento, la duración de los mercados y los plazos de las etapas subsiguientes.

ii. *Registro*: los agentes bróker registran en un directorio FIPA el rol que desempeñarán en cada mercado, que puede ser vendedor (subastador), comprador o ninguno si deciden no participar. Al tratarse de subastas inversas, el primer rol es propio de los consumidores y el segundo de los productores. Además, los agentes registran información específica a cada rol: los subastadores definen el precio máximo al que están dispuestos a comprar; y los compradores definen la cantidad máxima de carga que están dispuestos a suministrar.

iii. *Oferta*: los subastadores consultan el directorio FIPA para localizar productores e invitarlos a que pujen en sus subastas. Al final de esta etapa, todos los subastadores que no hayan encontrado suficiente oferta, así como todos los productores que no hayan encontrado subastadores, informan al ASPEM y cancelan su participación en el mercado. Estos nodos están obligados a aplicar la señal OpenADR que envió originalmente el operador del sistema.

iv. *Subasta*: entre todas las ofertas recibidas, los productores deciden en qué conjunto de subastas participar. Las ofertas se envían en forma de funciones lineales a trozos como la representada en la Figura C.9. Cada nivel de la función se corresponde con un nivel de consumo que el productor ofrece (*moderate*, *high*, *critical*). El número de secciones de la función depende del valor de la señal OpenADR de tipo *simple* que envía el operador, el cual marca el nivel de inicio.

v. *Resolución*: los consumidores deciden qué ofertas aceptar. El algoritmo implementado en este trabajo selecciona las ofertas con los precios más bajos. Para ello, en primer lugar, el algoritmo ordena todas las ofertas según el precio, entendiendo por *oferta* una sección de la función lineal a trozos enviada por el productor. A continuación, acepta ofertas de la lista de forma iterativa hasta cubrir toda la demanda. Por último, el consumidor informa a cada agente sobre el resultado de la subasta.

vi. *Cierre*: los ASPEM registran todos los acuerdos que han sido cerrados y, conforme a ellos, envía a cada agente bróker la señal OpenADR que este debe aplicar. Esta señal es transmitida a los agentes locales para que la apliquen sobre los recursos locales. En general, los productores que hayan logrado vender bloques de energía deberán adoptar niveles más restrictivos de consumo, y los consumidores que hayan cerrado acuerdos podrán mantener o incluso incrementar su demanda.

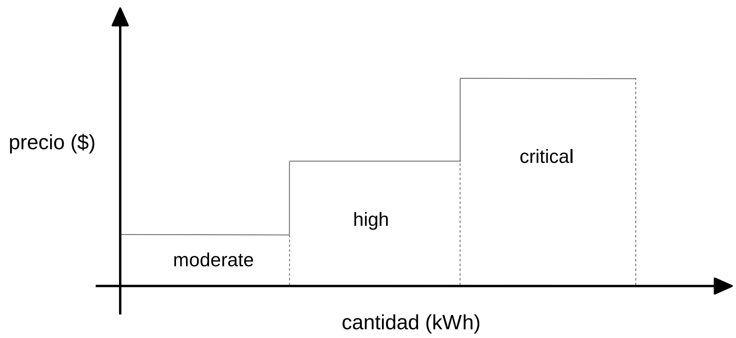

Figura C.9: Función lineal a trozos que envían los productores a los consumidores para representar sus ofertas.

La Figura C.10 ilustra el resultado de una simulación en la que el operador ordena una señal de tipo *moderate* que comienza a las 2:00 pm y termina a las 3:30 pm. Como resultado de los intercambios entre los nodos, el mercado genera una curva de demanda muy similar a la correspondiente al nivel *moderate*, siendo este el efecto esperado. Por su parte, la Figura C.11 muestra el resultado de la simulación para una señal de tipo *hard*. En ambos casos, el resultado que se obtiene usando mecanismos de mercado es seguro porque, cuando un subastador no logra suficientes ofertas para cubrir un nivel entero de demanda, este es obligado a cancelar la subasta y a aplicar la señal OpenADR de entrada. Con este comportamiento se logra que la demanda, como máximo, iguale el nivel de consumo ordenado por el operador.

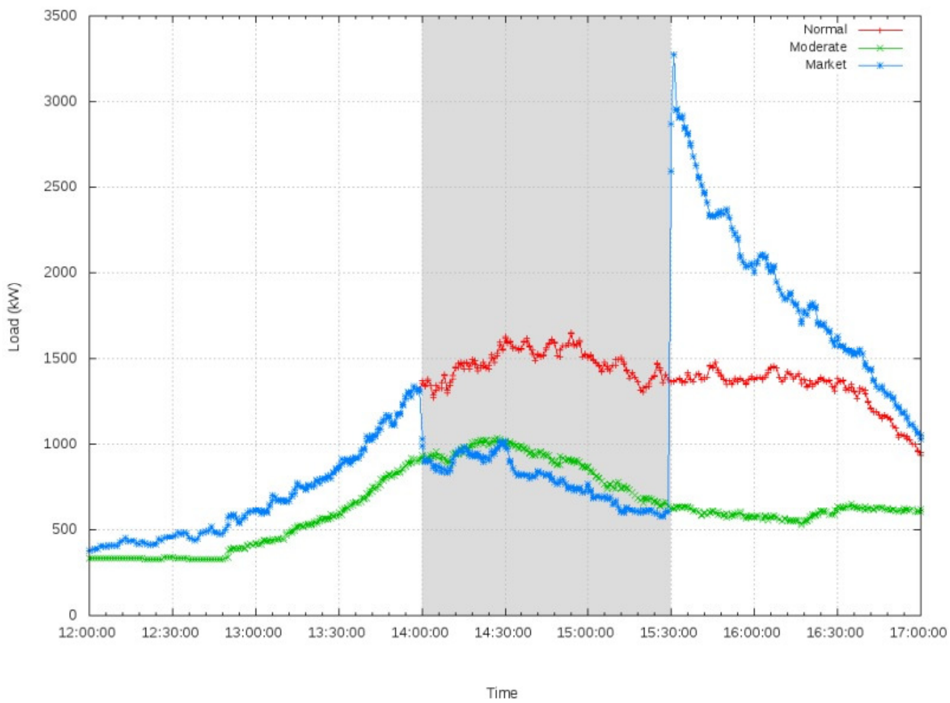

Figura C.10: Curvas de demanda cuando se aplica una señal OpenADR de tipo *moderate*.

La Tabla C.4 muestra los datos correspondientes a las simulaciones. La columna "*Cantidad intercamb.*" hace referencia al porcentaje de la cantidad subastada que, gracias a los intercambios del mercado, ha logrado ser cubierta. Como se puede observar, el porcentaje es significativamente mayor cuando no existen precios de entrada. Esto se debe a que, en este caso, las ofertas de todos los productores son válidas para todas las subastas. No obstante, el hecho más destacable es que,

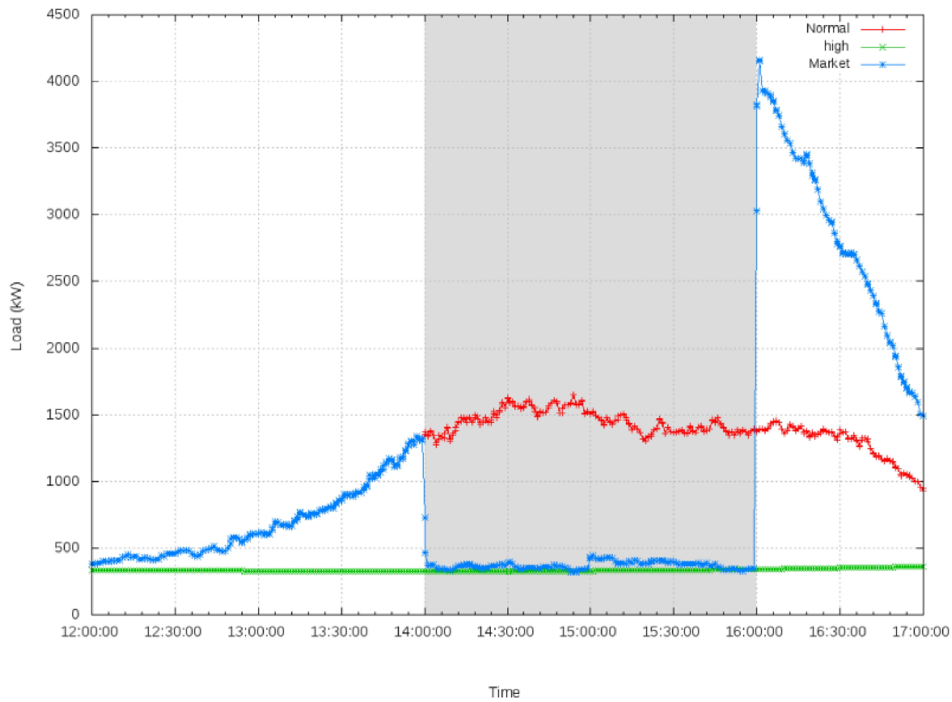

Figura C.11: Curvas de demanda cuando se aplica una señal OpenADR de tipo *hard*.

pese a que un número considerable de subastas se deben cancelar por no recibir suficientes ofertas, una cantidad importante de ofertas no logran participar en ningún intercambio. La razón para este hecho aparentemente contradictorio se halla en la mala distribución de las ofertas entre las subastas. En concreto, si muchos productores eligen participar en el mismo conjunto de subastas, muchas otras quedan excluidas de las negociaciones, no recibiendo suficientes ofertas para cubrir su demanda, y, por tanto, debiendo ser canceladas. En las simulaciones presentadas en este apartado los productores seleccionan las subastas de forma aleatoria. Bajo esta configuración, la mayor parte de las subastas reciben pujas, pero los factores R_{np} y R_{ca} no son lo suficientemente altos como para propiciar que todas las subastas reciban el número mínimo de ofertas de producción que requieren. La ineficiencia que causa la mala distribución de los participantes en los mercados de subastas paralelas se estudia en el siguiente apartado de este apéndice.

	Con precio de entrada			**Sin precio de entrada**		
	Cantidad intercamb. (kW)	**Subastas cubiertas (%)**	**Oferta vendida (%)**	**Cantidad intecamb. (kW)**	**Subastas cubiertas (%)**	**Oferta vendida (%)**
moderate	5798	37.97	18.06	12930	84.67	36.82
hard	3334	16.33	23.80	3843	18.82	20.70

Tabla C.4: Datos correspondientes a las simulaciones de los mercados de subastas paralelas cuando se aplican las señales *moderate* y *hard*.

El operador también puede enviar señales de tipo *delta*, las cuales en vez de ordenar la adopción de un nivel predefinido de consumo, indican la cantidad exacta de demanda que se debe descartar. Para gestionar esta clase de señales en la tesis se usa el parámetro *prioridad*, cuyo valor pueden acordar los usuarios con el ASPEM como parte del contrato. En este caso, el primer paso del procedimiento es crear una lista de las cargas compuesta de tres secciones. Estas, respectivamente, se corresponden con los tipos de carga *easy*, *normal* y *hard*. Las primeras dos secciones, tomando la prioridad como valor de referencia, se ordenan en forma descendente, mientras que la sección correspondiente a las cargas de tipo *hard* se ordena de forma ascendente. A continuación, se itera sobre la lista hasta cubrir la cantidad de demanda definida en la señal *delta*. En la Figura C.12 se muestra un caso en el que el operador ordena una reducción de 30.000 kW entre las 2:00 pm y las 3:30 pm. El objetivo de esta señal es evitar que los nodos deban adquirir el nivel *moderate*, el cual, en conjunto, resultaría más restrictivo para los usuarios.

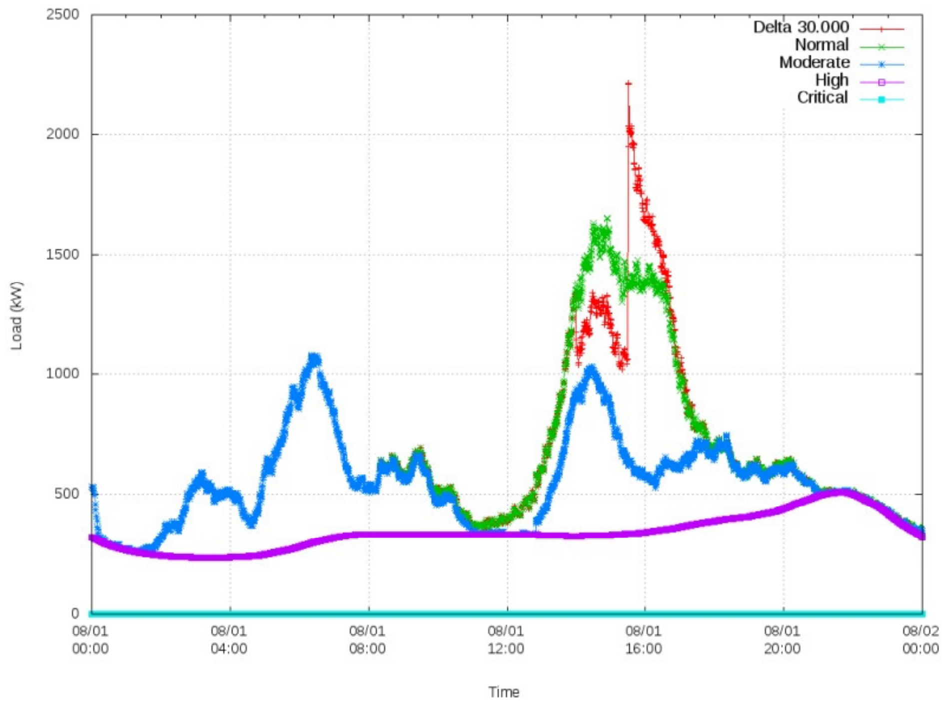

Figura C.12: Curvas de demanda cuando se aplica una señal OpenADR de tipo *delta* que ordena una reducción de 30.000 kW.

Los resultados demuestran que usando el modelo de Servicios de Agencia se pueden gestionar señales OpenADR a través de mercados. Esto, en contraposición con el esquema clásico, posibilita que los usuarios puedan participar activamente en el proceso de gestión y, con ello, defender sus intereses. Los experimentos también sirven para constatar que el modelo de Servicios de Agencia facilita que, sin perder capacidad de gestión, los nodos cliente sólo tengan que procesar señales de tipo *simple*, así simplificando notablemente su infraestructura.

C.3.8 Distribución de los compradores en mercados de subastas paralelas

En entornos grandes, como pueden ser el Smart Grid y las redes computacionales, un comprador puede recibir cientos o miles de invitaciones para participar en subastas, ya que los vendedores están interesados en incrementar la participación, y con ella la competitividad. Cuando existen más compradores que vendedores, la configuración ideal es que la participación de los primeros se distribuya de manera uniforme entre las subastas de los segundos. Sin embargo, esta condición no se puede esperar en entornos donde los compradores son agentes independientes, autónomos e interesados en sus propias metas. Además, a ello se ha de sumar que pueden existir factores objetivos que conduzcan a los compradores a preferir algunas subastas sobre otras. En particular, en este último caso cabe esperar que los compradores adopten **estrategias** a la hora de seleccionar las subastas en las que desean participar y que, además, estas estrategias sean compartidas, ocasionando así que los compradores se acaben concentrando en un pequeño grupo de subastas.

Para solventar este problema, en esta tesis se desarrolla el método HUDP (*Hash-based Uniform Distribution of Players*), el cual está inspirado en el funcionamiento de las tablas *hash*. El comportamiento básico de HUDP se resume en dos pasos:

i. Los compradores se registran en HUDP, de forma que a cada uno de ellos se le asigna un identificador.
ii. Cada comprador accede al HUDP para obtener la lista de subastas en las que puede participar.

La presencia de un mecanismo como HUDP puede alterar las condiciones de la competición y, por tanto, es necesario definir reglas que garanticen normas esenciales. En la tesis se proponen las siguientes cuatro *Prerrogativas*:

1. Si en la configuración original un comprador puede recibir suficientes invitaciones para alojar toda su capacidad, el mecanismo debe preservar esta condición.
2. Si en la configuración inicial un vendedor puede recibir pujas suficientes para vender toda su oferta, entonces el mecanismo debe preservar esta condición.
3. El mecanismo no puede actuar en perjuicio de un participante de forma deliberada.

4. El orden en el que los compradores acceden al mecanismo no debe afectar al resultado de las consultas.

Asimismo, el procedimiento de asignación se puede resumir en los siguientes puntos (Figura C.13): (*i*) crear una lista uniformemente distribuida de los vendedores; (*ii*) asociar a cada comprador una posición en la lista de vendedores; e (*iii*) ir asignando secuencialmente vendedores hasta cubrir, como mínimo, la capacidad de generación del comprador. A los compradores se les puede asociar más subastas de las que necesitan, haciendo posible así que puedan aplicar su propia selección sobre el subconjunto de subastas disponibles.

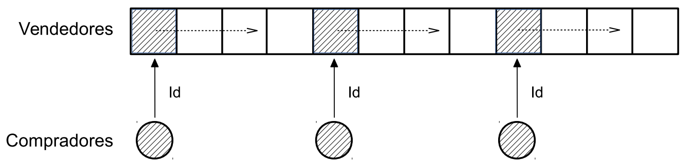

Figura C.13: Compradores asociados a una posición de la lista de vendedores.

El reto del procedimiento anterior es asignar posiciones de inicio a los compradores de forma que los vendedores reciban suficientes pujas como para poder asignar toda su oferta (*Prerrogativa 2*). Para averiguar el número de compradores que, por término medio, se debe asociar a cada vendedor ($nBpS$) se usa la cantidad media de producto ofrecida por subasta (U_s), la cantidad media que se ofrece por puja (U_b), y una constante (C_f) que permite corregir el número de compradores por subasta (habitualmente para incrementar el número y, con ello, la capacidad de selección de los vendedores).

$$nBpS = \left\lceil \frac{\overline{U_s}}{\overline{U_b}} \right\rceil + C_f \tag{C.1}$$

Para calcular de forma aproximada si existen suficientes ofertas de compra para cubrir todas las de demanda, se usa el parámetro α (Ecuación C.2), que es calculado en base al número de compradores (nB), el número de vendedores (nS) y el valor $nBpS$. Cuando:

- α es menor que 1, no existen suficientes compradores para cubrir todas las subastas.
- α es igual a 1, el número de compradores es suficiente para cubrir todas las subastas.
- α es mayor que 1, el valor de $nBpS$ se puede incrementar hasta $nBpS_c$ (Ecuación C.3).

$$\alpha = \left\lfloor \frac{nB/nBpS}{nS} \right\rfloor \tag{C.2}$$

$$nBpS_c = \left\lfloor \frac{nB}{nS} \right\rfloor \tag{C.3}$$

Si los identificadores asociados a los compradores se definen usando una secuencia de números naturales, entonces, cuando α es mayor 1, la posición de inicio (pos) de un comprador puede ser calculada conforme a la Ecuación C.4.

$$pos = \left\lfloor \frac{idBuyer}{nBpS} \right\rfloor \tag{C.4}$$

Cuando la cantidad de oferta de compra es inferior a la cantidad vendida ($\alpha < 1$), la única mejora que realmente se puede lograr es reducir el impacto negativo que puede causar la sobreconcentración de compradores cuando estos adoptan estrategias comunes. Para ello, en este caso se modifica el procedimiento anterior de forma que se asignan grupos de subastas a grupos de compradores. De este modo, si el contexto ofrece incentivos objetivos para preferir unas subastas sobre otras, la concentración de compradores queda acotada al grupo al que pertenecen. La Figura C.14 ilustra cómo la definición de grupos de vendedores y compradores limita la capacidad de concentración de estos últimos.

El número máximo de grupos de subastas (nS_g) que pueden ser creados depende del número mínimo de compradores que, por término medio, necesita una subasta. En concreto, se pueden crear nuevos grupos siempre que existan suficientes compradores como para cubrir al menos una subasta de los mismos. En general,

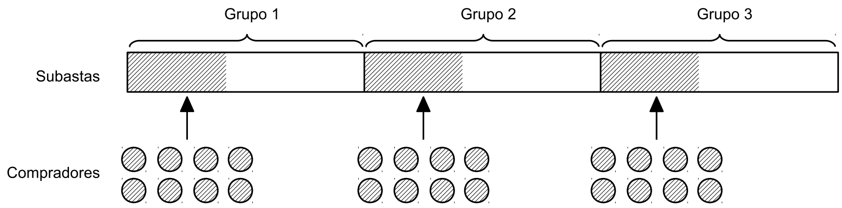

Figura C.14: La concentración de los compradores está limitada por la fronteras de los grupos de subastas.

mientras más grupos se creen, menor capacidad de concentración existirá, aunque también menos competitividad. Una vez se haya definido el número de grupos de subastas, se calcula cuál se le asigna a cada comprador usando la Ecuación C.7.

$$nS_g = \left\lfloor \frac{nB}{nBpS} \right\rfloor \tag{C.5}$$

$$nSpG = \left\lceil \frac{nS}{nS_g} \right\rceil \tag{C.6}$$

$$pos = \left\lfloor \frac{idBuyer}{nBpS} \right\rfloor * nSpG \tag{C.7}$$

Las ecuaciones definidas hasta el momento no contemplan la posibilidad de que las subastas puedan establecer precios de entrada, ni de que las ofertas se puedan definir a través de funciones lineales a trozos (Figura C.15). En la práctica, la presencia de estos factores impide que se pueda definir de forma precisa una distribución uniforme de los compradores porque: (*i*) la existencia de precios de entrada implica que no todas las pujas son válidas para todas las subastas; (*ii*) sólo algunas secciones de las pujas (de la función lineal a trozos) son aplicables; y (*iii*) el número de secciones por puja puede variar entre compradores. A continuación se define una solución basada en el enfoque anterior capaz de trabajar eficazmente en este contexto.

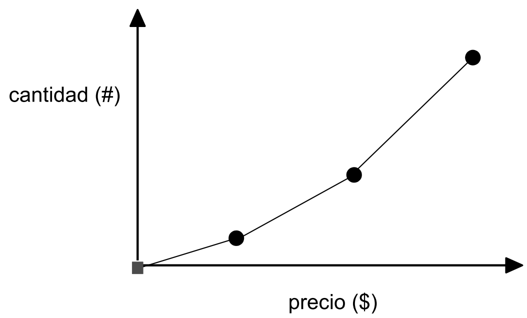

Figura C.15: Función lineal a trozos.

Cuando la capacidad de compra es mayor que la oferta, el parámetro $nBpS$ es calculado como la media del número de compradores por subasta (Ecuación C.8). Por su parte, el parámetro α se define igual que en el caso anterior (Ecuación C.2).

$$nBpS = \frac{\sum_i nBpS_i}{nS} \tag{C.8}$$

Como primer paso, el procedimiento ordena las subastas por precio de entrada en orden ascendente. Seguidamente, cada comprador es asociado a la primera subasta de la lista en la que puede participar. Esta condición se determina a partir del precio de entrada de la subasta y el valor del primer sector de la función lineal a trozos. Dado que la lista de subastas está ordenada, se cumple que, cuando una puja es asociada a una subasta (posición en la lista), dicha puja es válida para todas las subastas que siguen en la lista. Por su parte, cada subasta mantiene asimismo una lista de todos los compradores a los que se ha asignado esa subasta como posición inicial. Gracias a esta última estructura, una vez se conozca la subasta que corresponde inicialmente a cada comprador, para cada una de ellas se puede calcular el número de compradores extra que tiene asociados, tomando como referencia el parámetro $nBpS$. El número de compradores extra se acumula a lo largo de la lista de subastas. La Figura C.16 ilustra un ejemplo en el que $nBpS$ es igual a 2. En la posición de la subasta G existen dos compradores extra acumulados. Además, se aprecia que, en la subasta F, el número de compradores extra se decrementa porque esta subasta tiene asociado un número de compradores inferior al necesario (menor que $nBpS$).

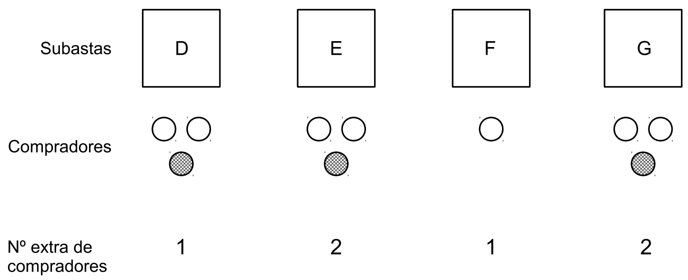

Figura C.16: Ejemplo de compradores extra acumulados en cada subasta.

El siguiente paso del procedimiento es mover los compradores que han sido clasificados como *extra* a subastas que tienen déficit de ellos. La Figura C.17 ilustra un ejemplo en el que $nBpS$ es igual a 2. Los elementos extra de las subastas H e I son reasignados a las subastas J y K. De este modo se busca que todas las subastas tengan asociados, como mínimo, un número de compradores igual a $nBpS$. En la práctica, las subastas con precios de entrada altos son pobladas con compradores que inicialmente fueron asociados a subastas con precios de entrada bajos.

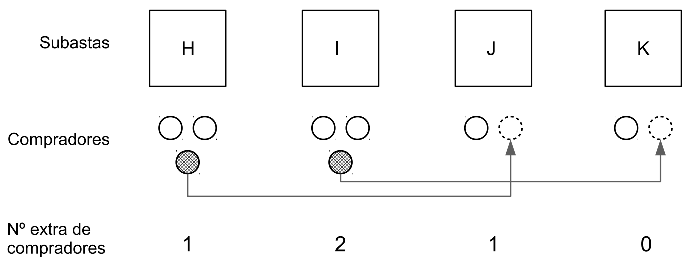

Figura C.17: Ejemplo de compradores que son reasignados a otras subastas.

Para dar soporte a las ofertas definidas a través de funciones lineales, el proceso de asignación de subastas se realiza para cada nivel de la función.

Por otra parte, cuando la capacidad de compra es inferior a la cantidad subastada, se procede igual que en el primer caso: asignando grupos de compradores a grupos de subastas. La diferencia es que, cuando existe precio de entrada, no es posible definir un grupo estático de subastas, sino una aproximación.

La tesis presenta en detalle los algoritmos que, conforme a las ecuaciones y procedimientos descritos, elaboran la lista de subastas que le corresponde a cada

Escenario	Subastado (kW)	Puja (kW)	R_{ca}	R_{np}
1	12653	36259	2.86	4.01
2	16211	40262	2.48	2.32
3	16937	31848	1.89	2.66
4	16753	26723	1.60	2.20
5	22789	25593	1.12	1.60
6	33602	25697	0.76	0.99

Tabla C.5: Descripción de los escenarios usados en los experimentos del mecanismo HUDP.

comprador. La mayor parte de estas operaciones, como se detalla en los algoritmos presentados en este documento, se realizan cuando el comprador se registra en el mecanismo HUDP, de forma que los accesos posteriores para obtener el listado de subastas son operaciones rápidas y poco costosas desde el punto de vista computacional.

Para evaluar los procedimientos propuestos se usaron los escenarios descritos en la Tabla C.5. Esta, además de la cantidad de energía subastada, incluye datos sobre dos factores importantes para entender la eficiencia del mecanismo de distribución. Estos son: (*i*) la relación entre la capacidad de compra y la cantidad subastada (en adelante R_{ca}); y (*ii*) la relación entre el número de compradores y el número de vendedores (en adelante R_{np}). Es necesario aclarar que los valores presentados en la tabla son el resultado de sumar los datos de tres mercados consecutivos de media hora que se instancian para gestionar un evento DR de 90 minutos. En todas las simulaciones el valor de la constante C_f es 0.

La Tabla C.6 muestra el resultado de la simulación para el *Escenario 1*, que destaca por tener mucha más capacidad de compra que de oferta. En este caso, cuando los compradores no disponen de incentivos para concentrarse, los resultados son positivos incluso cuando no se usa un mecanismo de distribución como HUDP. Ello se debe a que el número de compradores es alto y la participación

	sin HUDP		con HUDP	
	Aleatorio	**Estrategia**	**Aleatorio**	**Estrategia**
Intercambiado (kW)	10612	492	11672	11521
Oferta cubierta	84 %	4 %	93 %	91 %
Subastas canceladas	22	1	11	12
Subastas vacía	22	245	0	0

Tabla C.6: HUDP: Escenario #1: Resultados para los casos en los que no se usan precios de entrada.

tiende a seguir una distribución uniforme. Sin embargo, destaca que, pese a la gran capacidad de oferta, existe un número significativo de subastas vacías (subastas que no reciben ninguna oferta) y subastas canceladas (subastas que no reciben ofertas suficientes como para alojar toda la demanda subastada). Los resultados mejoran cuando se usa HUDP. En concreto, gracias a la mejor distribución de los compradores, no quedan subastas vacías y el número de subastas canceladas se reduce a la mitad. Por otra parte, cuando los compradores tienen incentivos para concentrarse, HUDP prácticamente logra anular su efecto; por el contrario, cuando HUDP no se usa y los compradores tienden a agruparse, el número de intercambios, y por tanto la eficiencia del sistema, se reduce drásticamente.

Como se muestra en la Tabla C.7, cuando las subastas establecen precio de entrada, el mecanismo de distribución también logra mejorar la eficiencia del sistema, anulando en gran medida la concentración de los compradores cuando estos adoptan estrategias comunes.

En el *Escenario 6* la capacidad de compra es inferior a la cantidad subastada, y asimismo el número de compradores es inferior al número de vendedores. Como se explicó, en estos casos la única mejora que se puede lograr es evitar la concentración de los compradores. Los datos de las tablas C.8 y C.9 demuestran que se logra este objetivo y que, además, cuando los compradores no tienen incentivos para agruparse, el mecanismo no causa ningún perjuicio.

	sin HUDP		con HUDP	
	Aleatorio	**Estrategia**	**Aleatorio**	**Estrategia**
Intercambiado (kW)	8493	950	9949	8727
Oferta cubierta	67 %	7 %	79 %	69 %
Subastas canceladas	31	8	33	49
Subastas vacías	41	222	1	4

Tabla C.7: HUDP: Escenario #1: Resultados para los casos en los que se usan precios de entrada y pujas en forma de funciones lineas a trozos.

	con HUDP		sin HUDP	
	Aleatorio	**Estrategia**	**Aleatorio**	**Estrategia**
Intercambiado (kW)	11663	487	11535	11565
Oferta cubierta	45 %	2 %	45 %	45 %
Subastas canceladas	249	3	285	164
Subastas vacías	179	739	177	271

Tabla C.8: HUDP: Escenario #6: Resultados para los casos en los que no se usan precios de entrada.

	con HUDP		sin HUDP	
	Random	**Strategy**	**Random**	**Strategy**
Intercambiado (kW)	10051	752	10824	10883
Oferta cubierta	39 %	3 %	42 %	42 %
Subastas canceladas	168	10	285	94
Subastas vacías	314	715	177	431

Tabla C.9: HUDP: Escenario #6: Resultados para los casos en los que se usan precios de entrada y pujas en forma de funciones lineas a trozos.

La Figura C.18 ilustra el rendimiento de HUDP en cada uno de los escenarios. La tendencia general es que el nivel de mejora logrado por HUDP se reduzca cuando también lo hace R_{ca}, ya que a menos participantes, menos posibilidades de lograr una mejor distribución de ellos. El aspecto más destacable es que el efecto de las estrategias es anulado en gran parte. Asimismo, otro efecto visible de HUDP es que resulta inocuo cuando los compradores no adoptan estrategias. El gráfico también muestra que, cuando no se adoptan estrategias y R_{ca} no tiene asociado un valor alto, no existe claro ganador. Esto se debe a que en este caso HUDP no tiene espacio para trabajar.

Para concluir, se puede afirmar que HUDP: (*i*) anula casi por completo la concentración de compradores cuando estos se guían por estrategias comunes; (*ii*) reduce el número de subastas canceladas; y (*iii*) resulta inocuo cuando no tiene posibilidad de actuar. Además, como se ha expuesto, el funcionamiento de HUDP se basa en operaciones sencillas e independientes, de forma que soporta accesos concurrentes.

C.4 Conclusiones

Muchas de las características que habitualmente se atribuyen al Smart Grid demandan la implantación de un sistema de gestión distribuido. Estos son proyectados de forma que los puntos de producción y consumo, representados por unidades de control inteligente, son capaces de planificar y negociar sus acciones directamente

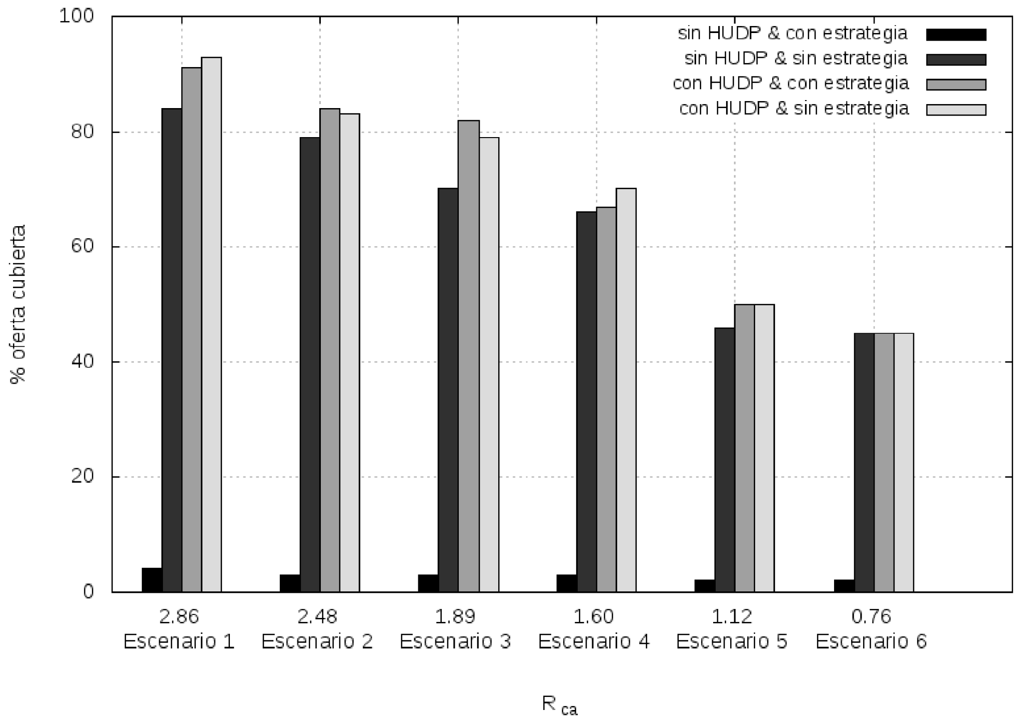

Figura C.18: Histograma de la cantidad de oferta cubierta cuando se usa el mecanismo HUDP y los participantes adoptan estrategias.

con el resto de entidades. En la práctica, la implementación de este modelo, conocido como SDM, adquiere la forma de un mercado de energía que se caracteriza por ser instanciado bajo demanda y de corta duración. Para hacer posible la participación autónoma y automatizada de los usuarios en esta clase de entornos se propone el uso de agentes inteligentes, los cuales, a priori, reúnen todas las características requeridas. Sin embargo, como se describe en este documento, los agentes inteligentes no han logrado el éxito esperado en entornos similares, haciendo necesario la elaboración de nuevos modelos de interacción y despliegue. Además, como también concluye este documento, responsabilizar a los dispositivos de control locales de tareas de negociación, coordinación y acceso a datos introduce retos estructurales y tecnológicos que contravienen muchas de las cualidades que se esperan del Smart Grid. Con el objetivo de superar esta barrera y de garantizar un sistema eléctrico reactivo, flexible y fiable, esta tesis, inspirándose en la experiencia obtenida en áreas similares, desarrolla el modelo de Servicios de Agencia. Este libera a los nodos locales de toda la complejidad que implica participar en sociedades virtuales y, a través de una solución inspirada en el paradigma Cloud Computing, logra conservar todas las propiedades que habitualmente se atribuyen a los agentes inteligentes. En este sentido, cabe destacar que el hecho de recurrir a la analogía de campos tecnológicos similares, identificando tanto las soluciones de éxito como los problemas que afrontan, ha constituido un recurso de gran utilidad para componer una solución nueva y práctica.

El modelo de Servicios de Agencia combina la orientación a servicios con los agentes inteligentes, constituyendo ambos una solución que supera muchos de los retos de interacción que presenta el Smart Grid. Por un lado, la orientación a servicios facilita el desarrollo de una red eléctrica en la que los clientes puedan contratar servicios conforme a sus necesidades; mientras que las cualidades de los agentes software, que el nuevo modelo conserva en su totalidad, proporcionan autonomía a los usuarios y, con ello, la capacidad de implementar sistemas de gestión distribuidos, reactivos e inteligentes.

El estado del arte de los algoritmos para la gestión de entornos propios del Smart Grid revela que características importantes de los mercados de energía son casi siempre omitidas. Entre ellas destaca la necesidad de aceptar ofertas combinadas de tipo complementario y suplementario. Sólo los algoritmos *CONSEC* y *mPJ* consideran esta funcionalidad; y sólo el último de los dos, basado en subastas paralelas inversas, proporciona un enfoque distribuido que explota todas las capacidades de los agentes autónomos. Esta tesis es el primer trabajo en simular el algoritmo *mPJ* en el entorno de las redes de energía, demostrando su validez. Asimismo es el primero en destacar el fuerte impacto que tiene la distribución de la participación de los compradores en la eficiencia de las subastas paralelas. Por otra parte, la revisión del estado del arte muestra que la mayor parte de los trabajos no están basados en estándares propios del Smart Grid, ni tampoco, en su mayoría, son reproducibles, lo cual afecta a la verificabilidad de las propuestas y, en consecuencia, a las garantías que ofrecen los mismos.

La infraestructura de simulación basada en GridLAB-D y Jade ha demostrado ser efectiva, siendo capaz de simular en detalle ambos campos de aplicación: el de las redes de energía y el de los agentes inteligentes. Además, el uso de interfaces bien definidas para la intercomunicación y sincronización de las dos herramientas de simulación ha probado generar un diseño limpio y escalable, fortaleciendo así la apuesta por soluciones de simulación conjunta. Del mismo modo, se ha de destacar que la sincronización de ambos simuladores, la cual es reconocida como la tarea más compleja en esta clase de soluciones, se vio simplificada por el hecho de que los dos proyectos fueran de código abierto.

Como demuestran los experimentos, los cuales están especialmente diseñados para ser realistas y reproducibles, el modelo de Servicios de Agencia es capaz de instanciar mercados de energía en programas DR. En estos entornos, que en principio están pensados para operar en base a comportamientos preprogramados, la presencia de nodos ASPEM, que incluyen entornos virtuales de negociación para los agentes software, demuestra: proporcionar autonomía a los usuarios, simplificar la infraestructura necesaria en las instalaciones del cliente, y facilitar la implementación de mercados guiados por los intereses de los usuarios. Asimismo, los resultados de los experimentos demuestran que, usando mercados basados en subastas paralelas y agentes software autónomos, se puede lograr la curvas de demanda correspondientes a las señales DR de entrada. Además, el modelo de Servicios de Agencia logra completar todos sus objetivos respetando los estándares del Smart Grid desarrollados por OASIS y NIST.

La concentración de compradores es una condición que puede afectar severamente a la efectividad de los mecanismos basados en subastas paralelas. Los experimentos demuestran que, cuando surge esta condición, el sistema puede tornarse inoperativo, invalidando completamente el uso de subastas paralelas, y con ello uno de los mecanismos más completos para implementar mercados en sistemas distribuidos como las redes computacionales o el Smart Grid. Esta tesis es el primer trabajo en estudiar en profundidad este efecto y en proponer una solución al problema. En concreto, el mecanismo HUDP, inspirado por el funcionamiento de las tablas *hash*, logra distribuir los compradores de manera uniforme entre las subastas sin alterar reglas básicas de los mercados. Además, es una solución especialmente diseñada para entornos concurrentes, distribuidos y reactivos, como el Smart Grid.

Para concluir, se ha de destacar que todo el trabajo desarrollado a lo largo de esta tesis cumple con los cánones de la *investigación reproducible*. El propósito de ello es que las contribuciones puedan ser contrastadas y, en caso de interés, extendidas por otros grupos de investigación.

Bibliography

[AB00] M. Amin and D. Ballard. Defining new markets for intelligent agents. *IT Professional*, 2(4):29–35, 2000. 31, 79, 89

[ACC+99] Gul A. Agha, Kathleen M. Carley, Jose Cuena, Edmund H. Durfee, Clarence Ellis, Les Gasser, Michael P. Georgeff, Michael N. Huhns andToru Ishida, Nadeem Jamali, Sascha Ossowski, H. Van Dyke Parunak, Anand S. Rao, Tuomas W. Sandholm, Sandip Sen, Munindar P. Singh, Larry M. Stephens, Gerard Tel, Jacques Wainer, Gerhard Weiss, Michael J. Wooldridge, and Makoto Yokoo. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999. 51

[All10] Robert John Allan. Survey of agent based modelling and simulation tools. Technical report, Science and Technology Facilities Council (STFC), 2010. 98

[Arn00] Tim Arnheiter. Modeling and simulation of an agent-based decentralized two-commodity power market. *Multi-Agent Systems, International Conference on*, 0:361–362, 2000. 31, 80

[ARP+02] C. Andrieu, B. Raison, D. Penkov, M. Fontela, S. Bacha, and N. Hadjsaid. Fault detection, analysis and diagnostics in high-dg distribution systems. Technical report, CRISP, 2002. 25

[ATS04] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys (CSUR)*, 36(4):335–371, 2004. 41

[BAV05] Rajkumar Buyya, David Abramson, and Srikumar Venugopal. The grid economy. *Proceedings of the IEEE*, 93(3):698–714, 2005. 44

[BBCP05] Fabio Bellifemine, Federico Bergenti, Giovanni Caire, and Agostino Poggi. Jade — a java agent development framework. In RafaelH. Bordini, Mehdi Dastani, Jürgen Dix, and Amal Fallah Seghrouchni, editors, *Multi-Agent Programming*, volume 15 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pages 125–147. Springer US, 2005. 98

[BBD+06] Rafael H Bordini, Lars Braubach, Mehdi Dastani, Amal El Fallah-Seghrouchni, Jorge J Gomez-Sanz, Joao Leite, Gregory MP O'Hare, Alexander Pokahr, and Alessandro Ricci. A survey of programming languages and platforms for multi-agent systems. *Informatica (Slovenia)*, 30(1):33–44, 2006. 98

[BCG+98] F. Brazier, F. Cornelissen, R. Gustavsson, C. M. Jonker, O. Lindeberg, B. Polak, and J. Treur. Agents negotiating for load balancing of electricity use. In *The 18th International Conference on Distributed Computing Systems*, page 622. IEEE Computer Society, 1998. 31, 87

[Bea04] David Booth and et al. Web Service Architecture, Feb 2004. 62

[Bra97] Jeffrey M. Bradshaw. *Software Agents*. AAAI Press, 1997. 51

[CA07] P. Carlsson and A. Andersson. A flexible model for tree-structured multi-commodity markets. *Electronic Commerce Research*, 7:69–88, 2007. 82, 88

[Chr99] Rich Christie. Power systems test case archive. `http://www.ee.washington.edu/research/pstca`, 1999. 97, 123

[CL07] Manish Chhabra and Hongen Lu. Towards Agent Based Web Service. In *6th IEEE/ACIS International Conference on Computer and Information Science*, 2007. 62

[CM06] Camarinha-Matos. *Virtual Organizations*, chapter ICT Infrastructures for VO, pages 83–103. Springer US, 2006. 46

[CMA03] Luis M Camarinha-Matos and Hamideh Afsarmanesh. Elements of a base ve infrastructure. *Computers in industry*, 51(2):139–163, 2003. 46

[CPPS06] G. Celli, F. Pilo, G. Pisano, and G. Soma. Optimal participation of a microgrid to the energy market with an intelligent ems. In *International Power Engineering Conference*, pages 663–668, 2006. 28

[CRI02] CRISP. Case study simulations and results. Technical report, Distributed Intelligence in Critical Infrastructures for Sustainable Power, 2002. 26, 83

[CSG08] DP Chassin, K Schneider, and C Gerkensmeyer. GridLAB-D: An open-source power systems modeling and simulation environment. In *Transmission and Distribution Conference and Exposition, 2008. T&D. IEEE/PES*. IEEE, 2008. 97

[cur15] curl. curl and libcurl. `http://curl.haxx.se`, 2015. 111

[DH04] A. L. Dimeas and N. Hatziargyriou. A multiagent system for microgrids. In *IEEE Power Engineering Society General Meeting*, number 55-58, 2004. 31, 77, 89

[DH05] A. Dimeas and N. Hatziargyriou. Operation of a multiagent system for microgrid control. *IEEE Transactions on Power Systems*, 20:1447–1455, 2005. 27, 31, 84, 85

[DJ03] Viet Dung Dang and Nicholas R. Jennings. Optimal clearing algorithms for multi-unit single-item and multi-unit combinatorial auctions with demand/supply function bidding. In *Proceedings of the 5th International Conference on Electronic Commerce*, ICEC '03, pages 25–30, New York, NY, USA, 2003. ACM. 74, 79, 160

[DOE09] DOE. Smart grid system report. Technical report, U.S. Department of Energy, 2009. 23

[Dra09] Nicola Dragoni. Where are all the agents? on the gap between theory and practice of agent-based referral networks. In *Principles of Practice in Multi-Agent Systems*, volume 5925 of *Lecture Notes in Computer Science*, pages 508–515. Springer Berlin Heidelberg, 2009. 51

[Ecl] Eclipse Foundation. Jetty. `http://eclipse.org/jetty`. 98

[ECN06] ECN. Crisp. distributed intelligence in critical infrastructures for sustainable power. http://www.crisp.ecn.nl/, 2006. 25, 81

[ETP06] Smart grids european technology platform. http://www.smartgrids.eu/, 2006. 23

[EU13] EU. *EU Energy, transport and GHG emissions, trends to 2050*. European Commission, 2013. 15

[EU14] EU. *EU energy in figures*. European Commission, 2014. 15

[FC09] Sergey Fomel and Jon F Claerbout. Reproducible research. *Computing in Science & Engineering*, 11(1):5–7, 2009. 90, 96

[FCD+13] J.C. Fuller, S. Ciraci, J.A. Daily, A.R. Fisher, and M. Hauer. Communication simulations for power system applications. In *Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), 2013 Workshop on*, pages 1–6, May 2013. 97

[FIP96] FIPA. FIPA - Foundation for Intelligent Physical Agents. `http://www.fipa.org`, 1996. 30, 98, 109, 154

[FIP01] FIPA TC Communication. FIPA Inform Communicative Act Specification. `http://www.fipa.org/specs/fipa00048`, 2001. 110

[FIP02a] FIPA TC Communication. FIPA Brokering Interaction Protocol Specification. http://www.fipa.org/specs/fipa00033, 2002. 64

[FIP02b] FIPA TC Communication. FIPA Query Interaction Protocol Specification. http://www.fipa.org/specs/fipa00027, 2002. 110, 178

[FIP02c] FIPA TC Communication. FIPA Request Interaction Protocol Specification. http://www.fipa.org/specs/fipa00026, 2002. 110, 178

[FIP03a] FIPA TC Communication. FIPA Agent Management Specification. http://www.fipa.org/specs/fipa00023, 2003. 109, 132, 148, 154

[FIP03b] FIPA TC Communication. FIPA Contract Net Interaction Protocol Specification. http://www.fipa.org/specs/fipa00029, 2003. 85

[FJK04] Ian Foster, Nicholas R Jennings, and Carl Kesselman. Brain meets brawn: Why grid and agents need each other. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 8–15. IEEE Computer Society, 2004. 44, 52

[FK97] Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of High Performance Computing Applications*, 11(2):115–128, 1997. 43

[FKT01] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222, 2001. 43

[FKT04] Ian Foster, Carl Kesselman, and Steven Tuecke. *The Grid 2: Blueprint for a new computing infrastructure*, chapter The Open Grid Services Architecture, pages 215–257. Morgan Kaufmann, 2004. 44

[Fos05] Ian Foster. Globus toolkit version 4: Software for service-oriented systems. In Hai Jin, Daniel Reed, and Wenbin Jiang, editors, *Network and Parallel Computing*, volume 3779 of *Lecture Notes in Computer Science*, pages 2–13. Springer Berlin Heidelberg, 2005. 43

[FR93] Daniel Friedman and John Rust. *The double auction market: institutions, theories, and evidence*, volume 14. Westview Press, 1993. 77

[FTNY08] T. Funabashi, T. Tanabe, T. Nagata, and R. Yokoyama. An autonomous agent for reliable operation of power market and systems including microgrids. In *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, pages 173–177, April 2008. 30, 84

[GC04] D. Greenwood and M. Calisti. An Automatic, Bi-Directional Service Integration Gateway. In *IEEE International Conference on Systems, Cybernetics and Man*, pages 10–13, October 2004. 62

[GDR04] Carole Goble and David De Roure. The semantic grid: Myth busting and bridge building. In *ECAI*, volume 16, page 1129, 2004. 43

[GLMS07] Dominic Greenwood, Margaret Lyell, Ashok Mallya, and Hiroki Suguri. The IEEE FIPA Approach to Integrating Software Agents and Web Services. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, pages 276:1–276:7, New York, NY, USA, 2007. ACM. 153

[GMD+10] T. Godfrey, S. Mullen, R.C. Dugan, C. Rodine, D.W. Griffith, and N. Golmie. Modeling smart grid applications with co-simulation. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 291–296, Oct 2010. 96

[GPA02] Miguel Gregori, Javier Palanca, and Gustavo Aranda. A Jabber-based Multiagent System Platform. In *Fifth Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1282–1284, 2002. 59

[Gro96] The PostgreSQL Global Development Group. PosgreSQL. `http://www.postgresql.org/`, 1996. 99

[GWTLT97] Andrew S. Grimshaw, Wm. A. Wulf, and CORPORATE The Legion Team. The legion vision of a worldwide virtual computer. *Commun. ACM*, 40(1):39–45, January 1997. 43

[HAIM07] Nikos Hatziargyriou, Hiroshi Asano, Reza Iravani, and Chris Marnay. Microgrids. *Power and Energy Magazine, IEEE*, 5(4):78–94, 2007. 27

[HDlPJ+12] Stella Heras, Fernando De la Prieta, Vicente Julian, Sara Rodríguez, Vicente Botti, Javier Bajo, and JuanM. Corchado. Agreement technologies and their use in cloud computing environments. *Progress in Artificial Intelligence*, 1(4):277–290, 2012. 187

[HdlPR+12] Stella Heras, Fernando de la Prieta, Sara Rodríguez, Javier Bajo, Vicente J. Botti, and Vicente Julián. The role of argumentation on the future internet: Reaching agreements on clouds. In *Proceedings of the First International Conference on Agreement Technologies, AT*, pages 393–407, 2012. 187

[HDT+05] N.D. Hatziargyriou, A. Dimeas, A.G. Tsikalakis, J.A.P. Lopes, G. Karniotakis, and J. Oyarzabal. Management of microgrids in market environment. In *Future Power Systems, International Conference on*, pages 7 pp.–7, Nov 2005. 30

[Hen07] James Hendler. Where are all the intelligent agents? *IEEE Intelligent Systems*, 22(3):2–3, 2007. 51

[Hol09] David G Holmberg. Facility interface to the smart grid. In *Gridop-Interop*, 2009. 32, 98

[Hop08] Tim Hoppe. An experimental analysis of parallel multiple auctions. FEMM Working Papers 08031, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management, 2008. 22, 89, 137, 147

[HTV+04] Nikos Hatziargyriou, Antonis Tsikalakis, J Vlachogiannis, K Papadogiannis, G Kariniotakis, JP Lopes, J Oyarzabal, C Moreira, A Madureira, and I Cobelo. Microgrid central controller strategies and algorithms, 2004. 27, 31

[IEA14] IEA. *World Energy Outlook 2014*. International Energy Agency, 2014. 2, 15

[Ign14] Ignite Realtime. Openfire. http://www.igniterealtime.org/projects/openfire/, 2014. 99

[ijl14] https://gitlab.com/groups/ijlopez-phd, 2014. 99

[ILR10] M. Hernandez-Tejera I. Lopez-Rodriguez. Agency services: An agent-based and services-oriented model for building large virtual communities. In *International Conference on Agents and Artificial Intelligence (ICAART)*, pages 309–314, 2010. 187

[Jia06] Z. Jiang. Agent-based control framework for distributed energy resources microgrids. In *IAT '06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 646–652, Washington, DC, USA, 2006. IEEE Computer Society. 31, 86

[KB15] Kalliopi Kravari and Nick Bassiliades. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11, 2015. 98

[KCKA04] Rene Kamphuis, Per Carlsson, J Kester, and H Akkermans. Market-oriented online supply-demand matching. Technical report, EU CRISP project, 2004. 28

[KDAP12] George Kyriakarakos, Anastasios I. Dounis, Konstantinos G. Arvanitis, and George Papadakis. A fuzzy logic energy management system for polygeneration microgrids. *Renewable Energy*, 41(0):315 – 327, 2012. 28

[KK09] Hak-Man Kim and T. Kinoshita. Multiagent system for microgrid operation based on power market environment. In *Telecommunications Energy Conference, 2009. INTELEC 2009. 31st International*, pages 1–5, Oct 2009. 86

[KND13] H.S.V.S. Kumar Nunna and S. Doolla. Multiagent-based distributed-energy-resource management for intelligent microgrids. *Industrial Electronics, IEEE Transactions on*, 60(4):1678–1687, April 2013. 85

[KS01] M. Klusch and K. Sycara. Brokering and matchmaking for coordination of agent societies: a survey. *Coordination of Internet agents: models, technologies, and applications*, pages 197–224, 2001. 63, 86

[KSLK03] J.D. Kueck, R.H. Staunton, S.D. Labinov, and B.J. Kirby. Microgrid energy management system. Technical report, Consortium for Electric Reliability Technology Solutions, 2003. 26

[KWK05] K. Kok, C. Warmer, and R. Kamphuis. The powermatcher: Multiagent control of electricity demand and supply. In *AAMAS*, pages 75–82, 2005. 30, 81

[LAH11] Vincenzo Liberatore and A Al-Hammouri. Smart grid communication and co-simulation. In *Energytech, 2011 IEEE*, pages 1–5. IEEE, 2011. 96

[Las02] Robert Lasseter. Integration of distributed energy resources. the certs microgrid concept. Technical report, Consortium for Electric Reliability Technology Solutions, 2002. 23

[LKG05] R. Lum, D.B. Kotak, and W.A. Gruver. Multi-agent coordination of distributed energy systems. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 3, pages 2584 – 2589, 10-12 2005. 31, 86

[LRHT10] I. Lopez-Rodriguez and M. Hernandez-Tejera. Agent-based services for building markets in distributed energy environments. In *Renewable Energy & Power Quality Journal (ICREPQ)*, 2010. 187

[LRHT11a] I. Lopez-Rodriguez and M. Hernandez-Tejera. Challenges of using smart local devices for the management of the smart grid. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 73–77, Oct 2011. 187

[LRHT11b] Ignacio Lopez-Rodriguez and Mario Hernandez-Tejera. Software agents as cloud computing services. In Yves Demazeau, Michal Pechoucek, JuanM. Corchado, and JavierBajo Pérez, editors, *Advances on Practical Applications of Agents and Multiagent Systems*, volume 88 of *Advances in Intelligent and Soft Computing*, pages 271–276. Springer Berlin Heidelberg, 2011. 187

[LRHT15] I. Lopez-Rodriguez and M. Hernandez-Tejera. Infrastructure based on supernodes and software agents for the implementation of energy markets in demand-response programs. *Applied Energy*, 158:1–11, 2015. xviii, 188, 189

[LRHTHC15] I. Lopez-Rodriguez, M. Hernandez-Tejera, and J. Hernandez-Cabrera. Regulation of the buyers' distribution in management systems based on simultaneous auctions and intelligent agents. *Expert Systems with Applications*, 42(21):8014 – 8026, 2015. xviii, 188, 190

[LRHTar] I. Lopez-Rodriguez and M. Hernandez-Tejera. Methods for the management of distributed energy environments using software agents and market mechanisms: A survey. *Electric Power Systems Research*, to appear. 188

[LSM09] J. Lagorse, M.G. Simoes, and A. Miraoui. A multiagent fuzzy-logic-based energy management of hybrid systems. *Industry Applications, IEEE Transactions on*, 45(6):2123–2129, Nov 2009. 28

[LSS+11] Hua Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili. Power system and communication network co-simulation for smart grid applications. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1–6, Jan 2011. 96

[LSW08] T. Logenthiran, D. Srinivasan, and D. Wong. Multi-agent coordination for der in microgrid. In *Sustainable Energy Technologies, 2008. ICSET 2008. IEEE International Conference on*, pages 77–82, Nov 2008. 80

[LXJM12] Martin Lévesque, Da Qian Xu, Géza Joós, and Martin Maier. Communications and power distribution network co-simulation for multidisciplinary smart grid experimentations. In *Proceedings of the 45th Annual Simulation Symposium*, ANSS '12, pages 2:1–2:7, San Diego, CA, USA, 2012. Society for Computer Simulation International. 96

[LZ14] Weilin Li and Xiaobin Zhang. Simulation of the smart grid communications: Challenges, techniques, and future trends. *Computers & Electrical Engineering*, 40(1):270 – 288, 2014. 40th-year commemorative issue. 97

[Mar98] Mauricio Marín. Asynchronous (time-warp) versus synchronous (event-horizon) simulation time advance in bsp. In David Pritchard and Jeff Reeve, editors, *Euro-Par'98 Parallel Processing*, volume 1470 of *Lecture Notes in Computer Science*, pages 897–905. Springer Berlin Heidelberg, 1998. 117

[MFPF01] Ma Ta Martinez, Philippe Fouletier, Ka Ha Park, and Joel Favrel. Virtual enterprise–organisation, evolution and control. *International journal of production economics*, 74(1):225–238, 2001. 45

[MG11] Peter Mell and Timothy Grance. The NIST definition of cloud computing. Technical report, National Institute of Standards and Technology (NIST), 2011. 53

[ML07] P. McBurney and M. Luck. The agents are all busy doing stuff! *Intelligent Systems, IEEE*, 22(4):6–7, July 2007. 51

[NEGR15] Nada A. Nabeeh, Haitham A. El-Ghareeb, and A. M. Riad. Integrating software agents and web services in service oriented architecture based cloud services discovery framework. *Journal of Cases on Information Technology (JCIT)*, 10:67–79, 2015. 187

[NER11] NERC. Balancing and frequency control. Technical report, North American Electric Reliability Corporation (NERC), 2011. 7

[NIS12] NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0, 2012. 32

[NT07] Mohammad Reza Nami and Djamshid Tavangarian. Virtual Organization: A New Approach in IT. In *IEEE International Conference on Computer and Information Technology (CIT)*, volume 1, pages 93–98, 2007. 45

[OAD] OpenADR Alliance. `http://www.openadr.org`. 11, 35

[OAD12] Openadr 2.0a profile specification, 2012. 35

[OASa] Energy interoperation version 1.0. 32

[OASb] Energy market information exchange (emix) version 1.0. 34

[OASc] Ws-calendar version 1.0. 34

[OJ05] J. Oyarzabal and J. Jimeno. Agent based micro grid management system. In *Future Power Systems (FPS)*, pages 16–18, 2005. 30

[Ora00] Oracle. Java Network Launching Protocol & API Specification (JSR-56), September 2000. 59

[PAG+09] Piette, Mary Ann, Girish Ghatikar, Sila Kiliccote, Ed Koch, Dan Hennage, Peter Palensky, and Charles McParland. Open Automated Demand Response Communications Specification (Version 1.0). Technical report, Lawrence Berkeley National Laboratory, 2009. 64

[Pal12] Mauricio Paletta. *Cloud Computing for Teaching and Learning: Strategies for Design and Implementation*, chapter Intelligent Clouds: By Means of Using Multi-Agent Systems Environments, pages 254–279. Information Science Reference, 2012. 187

[PAP] Requirements for smart meter upgradeability. 68

[PFR09] M. Pipattanasomporn, H. Feroze, and S. Rahman. Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference & Exposition (PSCE)*, 2009. 31, 87

[PJ05] Y. K. Penya and N.R. Jennings. Combinatorial markets for efficient electricity management. In *IEEE/WIC/ACM Int Conf on Intelligent Agent Technology*, pages 626–632, 2005. 79, 88, 89, 122, 131, 178

[PL06] Yoseba Penya Landaburu. *Optimal Allocation and Scheduling of Demand in Deregulated Energy Markets*. PhD thesis, Vienna University of Technology, 2006. 131

[PLSW06] L. Phillips, M. Link, R. Smith, and L. Weiland. Agent-based control of distributed infrastructure resources. Technical report, Sandia National Laboratories, 2006. 24, 31

[PS06] Michael Peters and Sergei Severinov. Internet auctions with many traders. *Journal of Economic Theory*, 130:220–245, 2006. 178

[Rez93] Larry Reznick. Using cron and crontab. *Sys Admin*, 2(4):29–32, 1993. 105

[RLS+14] S. Rohjans, S. Lehnhoff, S. Schutte, F. Andren, and T. Strasser. Requirements for smart grid simulation tools. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 1730–1736, June 2014. 97

[RPH98] Michael H. Rothkopf, Aleksandar Peke, and Ronald M. Harstad. Computationally manageable combinational auctions. *Management science*, 44(8):1131–1147, 1998. 76

[RPT07] S. Rahman, M. Pipattanasomporn, and Y. Teklu. Intelligent distributed autonomous power systems (idaps). In *IEEE PES Annual General Meeting*, pages 1–8, 2007. 31, 86

[RSEC11] B. Ramachandran, S.K. Srivastava, C.S. Edrington, and D.A. Cartes. An intelligent auction scheme for smart grid market using a hybrid immune algorithm. *Industrial Electronics, IEEE Transactions on*, 58(10):4603–4612, Oct 2011. 78

[RZ94] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. The MIT Press, 1994. 87

[San02] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, 2002. 52, 76, 178

[SBLH06] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, July 2006. 62

[Sch01] Rudiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. *Peer-to-Peer Computing, IEEE International Conference on*, 0:101–102, 2001. 41

[SJ06] Manuel Sanchez-Jimenez. Smart electricity networks based on large integration of renewable sources and distributed generation. Master's thesis, Kassel University, 2006. 6

[SS01] Tuomas Sandholm and Subhash Suri. Market clearability. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 1145–1151. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001. 74, 79, 160, 178

[Sta12] Tom Stanton. Are smart micrograms in your future? exploring challenges and opportunities for state public utility regulators. Technical report, National Regulatory Research Institute, 2012. 187

[Ste14] F. Bush Stephen. *Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid*, chapter Appendix: Smart Grid Simulation Tools, pages 489–492. John Wiley & Sons, Ltd, 2014. 97

[Sto09] Victoria Stodden. Enabling reproducible research: Licensing for scientific innovation. *International Jornal of Communications Law & Policy*, 13:1, 2009. 90, 96

[Str05] Marta Stryszowska. Last-minute and multiple bidding in simultaneous and overlapping second price internet auctions. CentER Working Paper Series, 2005. 178

[Tal12] D. Talia. Clouds meet agents: Toward intelligent cloud services. *Internet Computing, IEEE*, 16(2):78–81, March 2012. 187

[Tal14] Domenico Talia. Towards internet intelligent services based on cloud computing and multi-agents. In Salvatore Gaglio and Giuseppe Lo Re, editors, *Advances onto the Internet of Things*, volume 260 of *Advances in Intelligent Systems and Computing*, pages 271–283. Springer International Publishing, 2014. 187

[TTL05] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4):323–356, 2005. 43

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961. 79

[W3C] Soap version 1.2 part 0: Primer (second edition). 36

[WJ95] Michael Wooldridge and Nicholas Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10:115–152, 1995. 50

[Woo09] Michael Wooldridge. *An Introduction to MultiAgent Systems - Second Edition*. John Wiley & Sons, 2009. 51

[XMP] XMPP Standards Foundation. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. 35, 59, 99

[YA96] F. Ygge and J. M. Akkermans. Power load management as a computational market. In *Second International Conference on Multi-Agent Systems, ICMAS 1996, Kyoto, Japan*, pages 393–400. AAAI Press, 1996. 81

[YA00] F. Ygge and H. Akkermans. Resource-oriented multicommodity market algorithms. *Autonomous Agents and Multi-Agent Systems*, 3:53–71, 2000. 81

[YWM+05] Zhihong Ye, R Walling, N Miller, P Du, and K Nelson. *Facility Microgrids*. National Renewable Energy Laboratory, 2005. 87

[ZCB10] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1:7–18, 2010. 53

[ZK13] T.A. Mohamed Ziyad and Dr.K.R. Shankar Kumar. Agent based intelligent systems: General views from learning and research perspective. *Artificial Intelligent Systems and Machine Learning*, 5(4), 2013. 187

Declaration

I herewith declare that I have produced this work without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This work has not previously been presented in identical or similar form to any examination board.

The dissertation work was conducted under the supervision of Francisco Mario Hernández Tejera at the University of Las Palmas de Gran Canaria.

This page is intentionally left blank