Operation of a Multiagent System for Microgrid Control

Aris L. Dimeas, *Student Member, IEEE,* and Nikos D. Hatziargyriou, *Senior Member, IEEE*

***Abstract*—This paper presents the operation of a multiagent system (MAS) for the control of a Microgrid. The approach presented utilizes the advantages of using the MAS technology for controlling a Microgrid and a classical distributed algorithm based on the symmetrical assignment problem for the optimal energy exchange between the production units of the Microgrid and the local loads, as well the main grid.**

***Index Terms*—Auction algorithm, distributed generation, energy market, microgrids, multi agent system, symmetrical assignment problem.**

I. Introduction

A Microgrid is a possible future energy system paradigm [1], formed by the interconnection of small, modular generation (micro-turbines, fuel cells, PV, etc.), together with storage devices (flywheels, energy capacitors and batteries) and controllable loads at low voltage distribution systems. Such systems can be operated interconnected to the power grid, or islanded, if disconnected from the grid. The operation of micro-sources in the network introduces considerable complexity in the operation of an LV grid, but at the same time, it can provide distinct benefits to the overall system performance, if managed and coordinated efficiently.

Significant research is currently carried out regarding operation and control of Microgrids [2], [3]. In this paper, a control system is proposed, aimed at the following:

1) optimal use of local distributed resources;
2) feeding of local loads;
3) operation simplicity.

It should be noted that the optimal operation of the production units and the controllable loads does not refer exclusively to their market participation, but also to their technical operation, since it is not possible to have constantly specialized personnel for monitoring at this system level.

The paper presents a distributed control approach for Microgrids. These systems operate inside the wider distribution system and, for that reason, the overall environment is introduced first.

Three control levels are distinguished, as presented in Fig. 1.

- Distribution Network Operator (DNO) and Market Operator (MO) at the level of the Medium Voltage.
- Microgrid Central Controller (MGCC).
- Local Controllers (LC), which could be either Micro Source Controllers or Load Controllers.

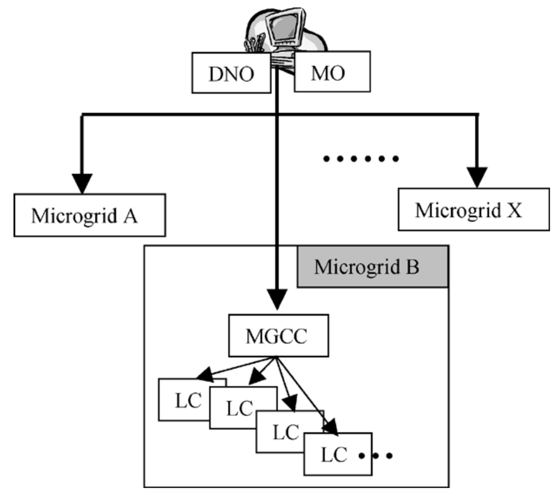

Fig. 1. Control levels of the microgrid environment.

The **DNO** is responsible for the technical operation in a medium and low voltage area, in which more than one Microgrids may exist. In addition, one or more **MOs** are responsible for the Market Operation of this area. These two entities do not belong to the Microgrid, but they are the delegates of the grid. The DNO refers to the operational functions of the system and the MO to the Market functions. It should be noted that, despite the autonomous operation of the Microgrid, it should ideally appear as a controlled, intelligent unit in coordination with the DNO. This is important in view of a possible increased number of such systems in the future.

The main interface between the DNS/MO and the Microgrid is the ***Microgrid Central Controller (MGCC).*** The ***MGCC*** is the main responsible for the optimization of the Microgrid operation, or alternatively, it simply coordinates the local controllers, which assume the main responsibility for this optimization.

The lower level of control consists of the **LC**. The LC's control the Distributed Energy Resources (DER), production and storage units, and some of the local loads. Depending on the mode of operation, they have certain level of intelligence, in order to take decisions locally. Of course, in any type of operation there are certain decisions that can be taken only locally. For example for voltage control, the LCs do not need the coordination of the MGCC and all necessary calculations are performed locally.

There are several levels of decentralization that can be possibly applied, ranging from centralized control to a fully decentralized approach. According to the fully decentralized ap-

Manuscript received November 8, 2004; revised March 14, 2005. This work was supported by the European Union research project "Microgrids" under Contract ENK 5-CT-2002-00610). Paper no. TPWRS-00590–2004.

The authors are with the National Technical University of Athens, Athens, GR 15773, Greece (e-mail: adimeas@power.ece.ntua.gr; nh @power.ece.ntua.gr).

Digital Object Identifier 10.1109/TPWRS.2005.852060

proach, adopted in this paper, the main responsibility is given to the DER controllers, which compete to maximize their production in order to satisfy the demand and probably provide the maximum possible export to the grid taking into account current market prices. Furthermore, LCs should take all the appropriate decisions to ensure safe and smooth operation of the DER they are controlling.

The organization of a controlled, intelligent entity (whole Microgrid) formed by several less intelligent entities (local controllers) can be based on a multiagent system (MAS). This was preliminary shown in a previous paper [4] of the authors. The multiagent technology is the evolution of the classical distributed technology with some specific characteristics that provide new capabilities in controlling complex systems [9]. This technology has already been proposed in the operation of large power systems, as well for the control of single machines [5]–[8].

The use of distributed control in a Microgrid provides effective solutions for a number of specific operational problems. For example, local loads and production or storage units may have different owners and several decisions should be taken locally and independently. Microgrids operating in a market require that the controller of each unit participating in the market, has a certain degree of intelligence. Moreover, the local DER not only sell power to the network, but may have also other tasks, like:

- producing heat for local installations;
- feeding local critical loads and preserving enough energy or fuel to supply them for some time;
- load-shedding capabilities;
- black start the microgrid;
- ensure a seamless transition from grid connected to island mode and vice versa;
- voltage control.

The general organization of the paper is as follows: In Section II the Market Operation of the Microgrid is presented. In Section III the Multi Agent Technology is shortly introduced together with some of its benefits for the control of a Microgrid. Section IV provides the mathematical background and the algorithm that ensure the optimal market operation of the system, as well some preliminary results in Section V. In Section VI, the method is adapted to the practical problem of the Microgrid Operation and Section VII provides a technical description of its MAS implementation. Section VIII describes the MAS installed in the laboratory Microgrid of the National Technical University of Athens. Section IX provides a comparison between the centralized and decentralized approach and Section X concludes.

II. Microgrid Market Operation

In order to analyze the control system presented in the next sections, it is necessary to describe the market model in which the Microgrid is assumed to operate. It should be noted that the system presented does not aim at the optimal participation of the Microgrid in the market, but at its optimal internal operation. According to the model adopted, optimal market operation is a task of the MGCC. The MGCC will make all the necessary negotiations with the Market Operator in order to achieve the best energy prices. The algorithm does not alter the demand of the local loads and the decisions of the MGCC, but performs optimal resource allocation. The DER units within the Microgrid adjust their set points, after negotiation with the other units based on the grid Market prices, their operational cost and the local load demands. It is also assumed that there is no limit to the power that can be sold or bought from the grid, since the capacity of the grid is almost infinite compared to the local demands.

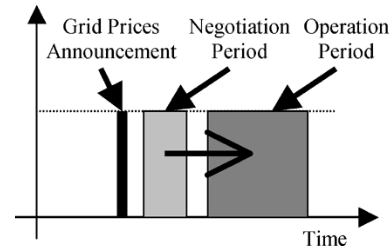

Fig. 2. Actions sequence for the market operation in the time domain.

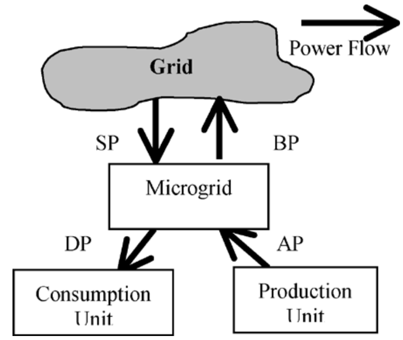

Fig. 3. Power flows and bids in the microgrid.

The overall procedure is the following (Fig. 2).

1. The Market Operator (MO) announces the prices for selling (SP) or buying (BP) energy to the Microgrid. Normally it is $\mathrm{SP} > \mathrm{BP}$.
2. The local loads announce their demands for the next 15 minutes and an initial price (DP) for the kWh. It is $\mathrm{DP} > \mathrm{BP}$ and $\mathrm{DP} < \mathrm{SP}$.
3. The production units accept or decline the load offer according to an Auction Price (AP) which is the result of the algorithm presented in Section IV (Fig. 3).
4. The negotiation continues for a specific time (5 min).
5. After the end of the negotiation time, all the units have adjusted their set points. If there is no production unit within the Microgrid to satisfy the load demand, the power is bought from the grid. In addition, the grid can be considered as a load too, so the production or storage units can sell energy to the grid.

For this market model, it is assumed that the distributed control system is an entity that tries to maximize the overall internal benefit, i.e., minimize the operational cost of the Microgrid.

It should be noted that the proposed control system with minor modifications can be easily adapted to other market models, for example, it can deal with longer term base negotiations for base load predictions. The horizon of the time scheduling is adaptable to the needs of the market model.

III. MAS Theoretical Background

Although there is no strict definition about what an agent is, the literature [9], [10] provides some basic characteristics:

The first characteristic is that an agent can be a physical entity that acts in the environment or a virtual one, i.e., with no physical existence. In our case the physical entity is the agent that controls directly a Microturbine and a virtual one a piece of software that makes bids to the energy market or stores data in a database.

An agent is capable of acting in the environment, i.e., the agent changes its environment with its actions. A diesel generator by altering its production changes the setpoints of the other local units, changes the voltage level of the adjacent buses and in a more global point of view changes the security level of the system [the stability of the system in case of a short circuit for example].

Agents communicate with each other and this could be regarded as part of their capability for acting in the environment. As an example, let's consider a system that includes a wind generator and a battery system: the battery system uses power from the wind turbine to charge and to discharge it, in times of no wind. In order to achieve this operation optimally, the two agents have to exchange many messages. This is a type of action because the environment is altered in a different way by this communication, rather than if the two agents were acting without any kind of coordination.

Agents have a certain level of autonomy, which means that they can take decisions without a central controller or commander. To achieve this, they are driven by a set of tendencies. For a battery system a tendency could be: "charge the batteries when the price for the kWh is low and the state of charge is low, too". Thus, the MAS decides when to start charging based on its own rules and goals and not by an external command. In addition, the autonomy of every agent is related to the resources that it possesses and uses. These resources could be the available fuel for a diesel generator.

Another significant characteristic of the agents is that they have partial or none at all representation of the environment. For example, in a power system the agent of a generator knows only the voltage level of its own bus and, maybe, it can estimate what is happening in certain specific buses. However, the agent does not know what is happening in the whole system. This is the core of the MAS technology, since the goal is to control a very complicated system with minimum data exchange and minimum computational demands.

Finally, another significant characteristic is that an agent has a certain behavior and tends to satisfy certain objectives using its resources, skills and services. An example of these skills could be the ability to produce or store power and an example for the services could be the ability to sell power in a market. The way that the agent uses the resources, skills and services characterizes its behavior. As a consequence, it is obvious that the behavior of every agent is formed by its goals. An agent that controls a battery system and its goal is to supply uninterruptible power to a load will have different behavior from a similar battery, whose primary goal is to maximize profits by bidding in the energy market.

IV. Mathematical Background

The core of the algorithm applied is based on an auction algorithm for the solution of the symmetric assignment problem. This method provides maximization of the internal benefit of the system. The symmetric assignment problem is formulated as follows.

Consider n persons and n objects that should be matched. There is a ***benefit*** a_{ij} for matching person i with object j. In the presented application, the benefit for each person is his revenues for obtaining object j, i.e., an agreement for producing a certain amount of energy. The main target is to assign the persons to objects and to maximize the ***total benefit***

$$\sum_{i=1}^{n} a_{ij}. \tag{1}$$

The ***price*** p is an algorithmic variable that is formed by the bids of all persons and so expresses the global desire. The prices of all objects form the price vector. These prices should not be confused with the market prices. Furthermore, the difference between the benefit and the price is the ***actual value*** of an object for a specific person. The actual value for a specific object is different for two persons, since it is related to the benefit. At the beginning of the iterations, the price vector is zero and so the actual value is equal to the benefit, although variations of the proposed methods use initial non- zero values for faster convergence.

In order to clarify the above terms, we consider an example with two objects and two persons that belong to a larger set of persons and objects. The first person has a benefit vector $\{\mathrm{a}_{11}, \mathrm{a}_{12}\} = \{10, 9\}$, the second one has benefits $\{\mathrm{a}_{21}, \mathrm{a}_{22}\} = \{7, 10\}$. Taking into account only the benefits, the first person has higher benefit for the first object and the second person for the second object. If we assume a price vector $\mathrm{p} = \{1, 8\}$ for the two objects, the actual values for the two persons are {9, 1} and {6,2}.Both players desire the first object more, since both have greater actual value for it than for the second, however the second person has greater benefit for the second object, than for the first. It can be said that the benefit represents local information for each person and the price vector global information for the whole system. The price for an object increases until at most one person wants it. Increasing the price of an object is an indication that there is another person that desires this object, too.

The auction algorithm used calculates the price vector p, in order to satisfy the ε-complementary slackness condition suggested in [11], [12]. The steps are described next.

At the beginning of each iteration, the ε-complementary slackness condition is checked for all pairs $(\mathrm{i}, \mathrm{j}_i)$ of the assignment. The j_i is the object j that person i wants to be assigned to. So the formulation of this condition is

$$a_{ij_i} - p_{j_i} \geq \max_{j \in A(i)} \{a_{ij} - p_j\} - \varepsilon. \tag{2}$$

$\mathrm{A(i)}$ is the set of objects that can be matched with person i.This inequality has two parts: $\alpha_{ij} - \mathrm{p}_j$ is the actual value of object j for person i , as described before. The right part refers to the object that gives maximum value to person i minus ε. ε is a positive scalar, added in the bid of each object, in order to avoid possible infinite iterations in case two or more objects provide maximum benefit to the same person, as explained later.

If all persons are assigned to objects, the algorithm terminates. Otherwise, a nonempty subset I of persons i that are unassigned is formed. Similarly, the nonempty subset $\mathrm{P(j)}$ is formed by the available objects. The following two steps are performed only for persons that belong to I.

The first step is the bidding phase, where each person finds an object j which provides maximal value; this is

$$j_i \in \max_{j \in A(i)} \{a_{ij} - p_j\}. \tag{3}$$

Following this, the person computes a bidding increment

$$\gamma_i = \mathrm{u}_i - \mathrm{w}_i + \varepsilon. \tag{4}$$

u_i is the best object value

$$u_i = \max_{j \in A(i)} \{a_{ij} - p_j\} \tag{5}$$

and w_i the second best object value

$$w_i = \max_{j \in A(i), j \neq j_i} \{a_{ij} - p_j\}. \tag{6}$$

According to the previous equations, the bidding increment is based on the two best objects for every person. The price of an object rises, if there are two or more bids for it and the price increment is the larger bidding increment between the bids. It is obvious that, if the scalar $\varepsilon = 0$ and the benefits for the first and the second best object are the same, then $\gamma_i = 0$ and this leads the algorithm to infinite iterations. The ε scalar ensures that the minimum increment for the bids is $\gamma_i = \varepsilon$.

The next phase is the assignment phase, where each object j selected as best object by the nonempty subset $\mathrm{P(j)}$ of persons in I, determines the highest bidder

$$i_j = \max_{i \in P(j)} \{\gamma_i\}. \tag{7}$$

Object j raises its prices by the highest bidding increment $\max_{i \in P(j)} \{\gamma_i\}$, and gets assigned to the highest bidder i_j. The person that was assigned to j at the beginning of the iteration, if any, becomes unassigned.

The algorithm iterates until all persons have an object assigned. It is proven [11] that the algorithm converges to the optimal solution, as long as there is one. The maximum number of iterations is

$$\frac{\max_{(i,j)} |a_{ij}|}{\varepsilon} \tag{8}$$

and the algorithm terminates in finite number of iterations if

$$\varepsilon < \frac{1}{n}. \tag{9}$$

The above algorithm is further explained by considering three persons and three objects. All bidders have zero benefit for the third object and the benefit for the rest of the objects is a constant $\mathrm{C} > 0$. So the benefits for the objects are $\mathrm{a}_{ij} = 0$ for i = 1,2,3 and $\mathrm{j} = 3$ and the other benefits $\mathrm{a}_{ij} = \mathrm{C} > 0$. The initial prices are considered zero. For this example, the first four iterations of the algorithm are presented in Table I.

TABLE I
RESULTS OF THE AUCTION ALGORITHM FOR THE FIRST FOUR ITERATIONS

Number of iteration	Prices	Pairs	Bidder/ Object	Incre- ment
1	0, 0, 0	(1,1)(2,1)(3,1)	3/1	ε
2	ε, 0, 0	(1,2)(2,2)(3,2)	2/2	2ε
3	ε, 2ε, 0	(1,1)(2,1)(3,1)	1/1	2ε
4	3ε, 2ε, 0	(1,2)(2,2)(3,2)	3/2	2ε

The second column of the table called “Prices” shows the price of the three objects at the beginning of each iteration. The column called “Pairs” shows the pairs (persons, object) that are assigned at the end of the iteration. The fourth column called “Bidder/ Object” shows which person was the bidder at the end of the negotiation and for which object a bid is made. Since we have only three objects and three persons, there can be only one bidder in each iteration. The last column shows the bidding increment at the end of the iteration.

In the first iteration, all persons desire object 1, so there are bids for this object. It should be mentioned that, in case the bids are equal, some selections are random or based on rules. The initial selection could be the second object or two persons could bid for the first object and the other for the second. The selected strategy does not affect the convergence of the algorithm according to [11]. According to the benefits, all the persons in this iteration have the same increment bid, the value of the bid is ε and the winning bidder is the last person. In the second iteration the bid increment for all persons is 2ε because

$$\gamma_i = (\mathrm{u}_i - \mathrm{w}_i) + \varepsilon = \varepsilon + \varepsilon = 2\varepsilon. \tag{10}$$

In this example, the need of ε is clearly illustrated, because otherwise no price would increase. All persons desire object 1 or 2, since the benefit is $\mathrm{C} > 0$ for both of them. In the last iteration the prices of 1 or 2 object will be greater than C, object 3 receives a bid and this is the end of the algorithm.

V. Preliminary Results

In order to explore the performance of the algorithm, a simple program was developed in a single processor machine using

TABLE II
RESULTS OF THE AUCTION ALGORITHM FOR VARIOUS SIZES OF THE PROBLEM

Number of objects	10	20	30	40
Average number of iterations	11	47	80	110
Max number of Iterations	15	80	105	250
Min number of Iterations	8	17	22	34

random values for the benefits of the objects. The primary concern was to discover the average and the maximum number of iterations needed to converge to solution. This is critical for this application, since in the distributed control system, there is significant delay in each iteration. During the development and testing of the Java program, it was concluded that it is not effective to have more than 100 iterations, since otherwise negotiations would last several minutes or hours. It should be noted that in the single machine application, there are no communication delays. In a multiagent application, built over a local network or the Internet, the communication delay should be seriously considered in the development. In Table II, results from the program are presented.

The conclusion from Table II is that no more than 30 objects (and 30 persons) should be used. This is enough for our application and detailed justification for this selection will be given in the next section. Furthermore, the use of proper value for the ε factor ensures that there will be no more than 150 iterations. This means that $\max |\mathrm{a}_{ij}|/\varepsilon < 150$.

VI. MICROGRID MARKET OPERATION EXPRESSED AS SYMMETRICAL ASSIGNMENT PROBLEM

In order to describe the Microgrid market operation, two types of physical agents and one type of virtual agent are introduced. The two physical agents are the Production Unit Agent and the Load Unit Agent. These two agents are physical, because they directly control a production or storage unit and a load panel, respectively. The third type is the Grid Agent. This agent is virtual, because it cannot control the grid in any way and just announces the prices for selling or buying energy. All other agents, introduced later in this section, are virtual and their operation concerns the auction algorithm only.

Let us consider that there are x production units with total capacity X and y Loads with total capacity Y. The symmetrical assignment problem requires that $\mathrm{X} = \mathrm{Y}$. In order to overcome the problem of surplus or deficient local production, a virtual load with proper price is added, as shown in Fig. 4. Similarly, virtual production can be added. The virtual load or production corresponds to the extra energy that is bought or sold to the grid. As mentioned before, it is assumed that the grid can offer or receive infinite energy.

In order to apply the algorithm for the solution of the symmetric assignment problem, the load should be divided into equal blocks, similar to the available production. Blocks that belong to the same load have equal benefits, since the system will provide all the necessary power for the whole load or none. For example, if we consider a water heater that demands 500 Wh for the next 15 min, the system should provide 500 Wh or nothing.

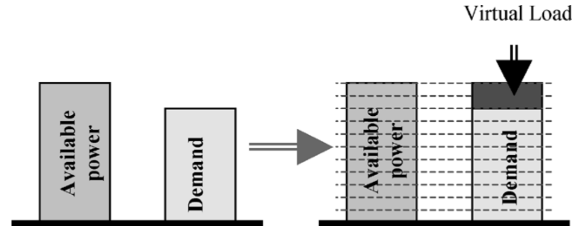

Fig. 4. Blocks of energy that form the assignment problem.

As explained in Section V, it is desirable to limit the algorithm execution to 100 iterations at each negotiation, i.e., a number of 20 to 30 blocks should be selected including the additional virtual load. According to the problem formulation of Section IV, the "persons" correspond to the blocks of the Available Power and the "objects" to the demand blocks. The agent market operation based on the described model is illustrated in Fig. 5. The Production Unit Agents control the DER, the Load Unit Agents represent the loads and the Grid Agent generates Market Player Agents. The Market Player Agents are virtual agents and their task is to accomplish the negotiation. There are two types of Market Player Agents: The Seller and the Buyer. The Buyer is the object in the assignment problem and the Seller is the person. Every Market Player Agent represents a single block of energy.

Similar to the local loads, the virtual load is represented by Market Player Agents that are created from the Grid Agent. According to the proposed market model, each producer has the ability to sell all the production to the grid and, similarly, every load can buy energy from the grid. For this reason the Grid Agent finds the number of pairs of Market Player Agents (Sellers and Buyers) and creates extra sellers and buyers. The number of the agents is equal to Market Agents that are created from the Production Units Agents and the Load Unit Agents. In this way, buying or selling energy from the grid is determined by the algorithm.

A major issue in Microgrids operation is the estimation of the upper limits for the demand or the available power of each DER for the next time interval. This should be done for each participant, separately. It should be noted, that although forecasting techniques are well advanced for larger interconnected systems and typically hourly resolution times, there is little experience in forecasting with a high temporal resolution (e.g., < 15 min) with a horizon of 3–4 h for very small loads, like the loads in a Microgrid. This happens because the load fluctuation in a small system is very high and has no specific pattern.

In this application the upper limit is defined by two methods depending on the type of load or DER. The first method is to consider that the upper limit is the nominal capacity. For units like a Diesel Generator or a water heater this is quite realistic. On the contrary, for units like Photovoltaic Panels, Wind generators or lights, the persistent method is used, i.e., it is assumed that the average energy production or demand for the next 15 min will be the same as the current one.

It should be noted that other functionalities of the Microgrid (like security check, battery management, voltage control, etc.) can be included in this operation. For example, the offered power of battery bank production could be reduced in order to

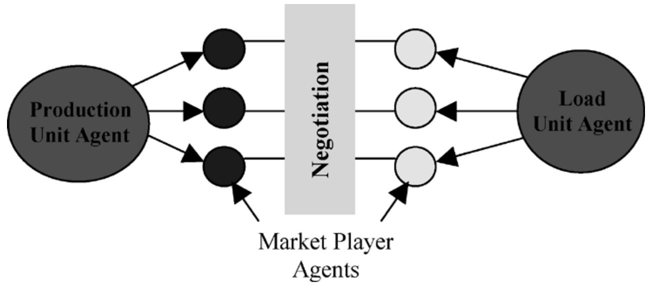

Fig. 5. Virtual market player agents that are created for the need of the negotiation.

maintain the state of charge status and keep certain amount of energy to serve the system, in case of a grid black out.

VII. MAS Implementation

The Microgrid is a micrograph of a large power system, encapsulating a large amount of technical parameters and details. Centralized control is possible in two ways: the first option is to create a very complex model with full description of all its elements and, as a consequence, requires a communication system capable to exchange large amount of information, as well as complex algorithms. The second option is to create a simplified description of the system and obtain results that would probably be suboptimal. The MAS technology allows creating a model of the system, as detailed as possible. In this model, every agent uses the exact piece of information it needs, leaving the technical details for the agents that are below it in the organization chart. It is therefore essential to have a formal way of describing information and giving to each agent the information it needs. In our application, the system model has three types of agents, as shown in Fig. 6.

- ***Control Agents***: they control directly the physical units of the system.
- ***Management Agents:*** they manage the Microgrid and take decisions regarding the state of it.
- ***Ancillary Agents***: they perform ancillary services, like communication tasks or data storage.

The development of the application is in conformity with the standards proposed by the international Foundation of Intelligent Physical Agents (FIPA) [13]. This organization aims to standardize the development of such systems, especially in the area of communication between the agents and the organization of the MAS. For our implementation, the Java Agent DEvelopment Framework (Jade) 3.0 platform is selected [14]. Jade is a Java-based tool for developing MAS systems.

One main feature of FIPA compliant MAS platforms is using the Agent Communication Language (ACL) [15] with high-level ontologies. It is a general conclusion that high-level communication is one significant element [9], [10] to develop intelligence inside a society, no matter if this society is a human one or a MAS. For the system presented here, an ontology was developed according to the needs of the system. This ontology supports two main tasks.

The first task is to provide an adequate description of the system, so that it includes all the necessary information for its control. Thus, all software modules of the control system perceive the system in the same way and this is important in order to develop a high level communication system, where the agents exchange knowledge. In order to understand the importance of having a common perception of the environment, consider the concept "energy" which has different meaning in quantum physics, food, or electric systems. Furthermore, the object oriented nature of the ontology and the data abstraction, support the development of a distributed control system, since each agent handles only the necessary (or allowable) part of information and knowledge. Fig. 7 presents a small simplified part of the ontology concerning the loads, which shows the advantages of this architecture. The main class Ont_load provides general details about the whole load bank and the inherited classes have details for each sub load, like market values or the address in the Programmable Logic Controller (PLC) that controls the switch of the specific sub load. The load unit agent needs to know and handle only the information of the ont_load class and nothing more. Furthermore, the Market Player agent needs to know the information of the ont_market class. This architecture simplifies extremely much the algorithmic development of each function of the system.

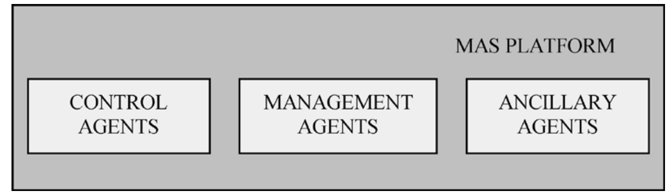

Fig. 6. Three types of agents for the proposed MAS.

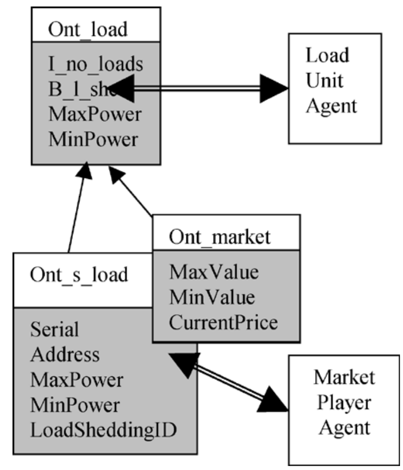

Fig. 7. Part of the proposed ontology for the MAS.

The second main task is to support high level communication, so that the agents not only exchange simple values, but also knowledge, commands, beliefs or procedures that have to be followed. For example, the agent that controls a load participates in the market by sending a request message to all the production units agents declaring the amount of energy that it needs. In ACL, this message has the following parts.

1) **Message Type**: in our case is a request.
2) **Receiver**: All the agents that produce energy and participate in the market.

3) **Language**: The agents should use the same language in order to parse the messages correctly.
4) **Ontology**: The agents share the same ontology in order to understand the content of the message.
5) **Content**: in our case is an object of the ontology called energy.

The agents that participate in the market communicate by sending proposals and accepting or rejecting proposals and this is how the auction algorithm was implemented.

Using ontology and conversation based communication, MAS can relatively easily deal with other functionalities of the system, provided that the agents understand a larger ontology and have the ability (intelligence) to handle a larger variety of messages. For example, if they receive a message that a black out occurred, they understand that it is normal to measure zero voltage and that they should stop their participation in the market. After that, they should follow some predefined actions, in order to restore the system in isolated mode and feed at least some critical loads. In a similar way, the functionalities mentioned in the introduction can be implemented in the control software.

The final issue is how agent behaviors are implemented in the MAS software. The JADE platform provides an excellent library for implementing behaviors by using a basic object called Behavior. Furthermore the platform provides a number of classes that extend the basic object Behavior, such as:

1) SimpleBehavior;
2) OneShotBehavior;
3) CyclicBehavior;
4) CompositeBehavior;
5) SequentialBehavior.

The JADE provides as well methods for manipulating these behaviors. Detailed description of this functionality can be found in [16]. Using these classes, next to auxiliary behaviors for message handling and monitoring, three main behaviors have been introduced:

1) Market Operation;
2) System Stand By;
3) System Start Up.

The first behavior refers to market participation, the second does nothing (stand by mode) and the third performs an elementary system start up with predefined critical loads.

The implementation of each behavior is not the same in every agent, since the main type of agents (Production Unit Agents, Load Units Agents, and Market Player Agents) have different roles in the system. Furthermore, the Market Player Agents do not have the System Start Up behavior, since they are virtual agents created for the market operation only. One critical point is that the first and the last behaviors are incompatible and, in order to avoid dangerous situations, the MGCC coordinates the transition between the behaviors by sending proper messages. The simple but effective solution is to send all agents to stand by mode and then to select the new behavior. A detailed description of how these behaviors are implemented is beyond the scope of this paper.

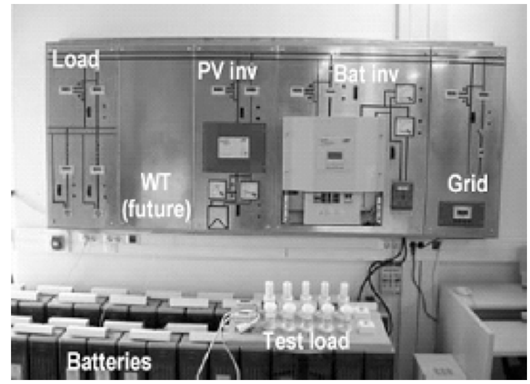

Fig. 8. Laboratory microgrid in the Power System Laboratory of the National Technical University of Athens.

VIII. MAS Implementation at a Laboratory Microgrid

In Fig. 8, the Microgrid installed in the Power System Laboratory of the National Technical University of Athens is shown. This is a modular system, comprising a PV generator as the primary source of power. The micro sources are interfaced to the 1-phase AC bus via DC/AC PWM inverters. A battery bank is included, interfaced to the AC system via a bi-directional PWM voltage source converter. The Microgrid is connected to the local LV grid [17]. Furthermore, a panel with controllable loads is available. This Microgrid is the test field of the MAS application.

In this section a general description of the part of the MAS that controls the laboratory system is presented. This part of the MAS incorporates the following agents.

- MGCC: a **Microgrid Central Controller Agent** that announces the beginning and the end of the negotiation period for the market operation.
- PV: a Production Unit Agent that is dedicated to represent the photovoltaic panel.
- Battery unit: a **Production Unit Agent** that is dedicated to represent the battery panel. This agent can sell or buy energy from the market depending on the state of charge of the batteries.
- Power system: a Grid Agent that represents the grid. It should be mentioned that the grid is considered as a production unit or consumption unit with infinite capabilities.
- Load unit: a **Load Unit Agent** that is specialized in representing the loads of the system.
- Power seller agents: **Market Player Agents** that bid directly in the market and represent the agents that sell power to the system. According to the assignment problem, these agents are the “persons”.
- Power buyer agent: **Market Player Agents** that participate in the market and represent the agents that buy power from the system. According to the assignment problem, these agents correspond to “objects”.

An operational cycle of the system is the following.

1) The MGCC announces the beginning of the market period.

2) The power sellers estimate their actual production capabilities and generate the proper number of power-seller market agents.
3) The loads estimate their actual demands and generate the proper number of power buyer market agents.
4) The power system agent, depending on the number of the seller and buyer agents, generates sufficient number of agents, in order to make the system symmetrical. Next, it calculates the total number N of the sellers (or buyers) and generates an extra set of N sellers and N buyers. This allows a production unit to sell directly to the grid or gives the ability to a load to buy directly from the grid.
5) The MGCC announces the start of the new negotiation cycle.
6) The energy market sellers and buyers agents bid in the market, according to the algorithm described in Section IV.
7) The MGCC announces the end of the current negotiation
8) The sellers and buyers agents announce to their parent agent their assignment and then terminate themselves.
9) The MGCC announces the end of the energy market period.
10) The MGCC announces the new set points

It is important to consider the required negotiation time according to the auction algorithm. Every negotiation cycle needs a maximum of eight messages between each pair of seller and buyer. There is a maximum delay of approximately 200 ms for the receipt of a message, i.e., each cycle needs approximately 2.5 s. This explains why the maximum number of iterations should be less than 100; otherwise each negotiation could last for hours.

A total number of 30 agents can therefore be implemented, which means that the power production or load units can produce 15 agents (considering that the power system unit generates an extra number of agents equal to the initial). If every energy block is 250 Wh, the system can handle 3750 Wh. In the application presented, the block size is fixed, but it can be easily adjusted, according to the total demand or available production.

A critical point of the application concerns communication, not only between the agents, but also between agents and the various devices. Several technologies have been used, like eXtentible Markup Language (XML), XML remote procedure call (XMLRPC), andObject linking and embedding for Process Control (OPC). An optimized usage of these technologies would minimize the delay up to 50 or 60 ms, in order to allow the deployment of more than 50 agents. This number is adequate for the control of a larger and more realistic Microgrid, like the one presented in Fig. 9 [18]. This network includes 6 DER units, including Photovoltaic panels, Fuel Cell, Batteries, Wind Generators and CHP based on Microturbines.

It should be noted that, due to space limitations, the approach presented here does not deal with other functionalities that the control mechanism of a Microgrid should have. Although not included in the presented implementation, two main categories of functionalities can be distinguished:

1) Energy Related Functionalities, like reserve power handling.

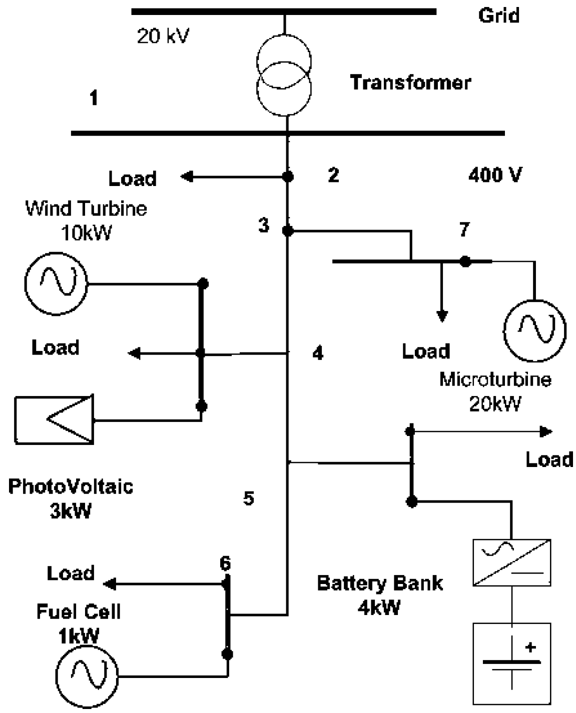

Fig. 9. Study case network of a microgrid.

2) Control Actions, like seamless transition to isolated mode or reconnection to the main grid

In the first category, each agent can calculate the amount of power needed for the specific operation and place a bid in the auction algorithm

In the second category, coordination algorithms should be included in the agents. An example is the reconnection to the main grid, where the system has to ensure that the main grid has acceptable voltage and frequency and, after that, it may proceed to all the necessary switching actions.

IX. Comparison of Decentralized Versus Centralized Approach

It is interesting to compare the decentralized approach proposed in this paper with a more centralized one, where the Microgrid Central Controller decides about the setpoints of each unit [19]. The main difference between the two approaches lies in the amount of information that is processed in each case. If the MGCC had available and could process all the information of the LCs, then its solution would be “at least as good”, as the one provided by the decentralized control. This is because each LC does not have direct access to the information of its neighbor controller, although MAS technology allows to ask for it. In practice however, it is very difficult for the MGCC to have access to all available information. For example, it is very complicated for the MGCC to know and handle the temperature of the battery of a specific storage unit. Similar conclusions can be found in the literature [20]. It is very hard to implement at a reasonable cost a centralized system that can bid in the market every hour and, at the same time, has the ability to shut down a specific load or change a specific generator set point, in case of unstable operation within the next 300 ms.

A related issue concerns data communication infrastructure. The proposed method only needs a simple local network and the information exchange is limited to the essential data only. A more centralized approach requires a significant data flow toward a single central point, in order to achieve similar results. The problem becomes extremely difficult and expensive to solve, if real time functionalities are required, such as on line security assessment.

Another issue is the openness of the system. Adopting a decentralized approach, allows every manufacturer of DER units or loads to embed a programmable agent in the controller of his equipment, according to some rules. This would provide the required "plug and play" capability of future DER units and loads. On the contrary, in a centralized system the installation of any new component would require extra programming of the central controller.

X. Conclusions

A Microgrid, although small in size, has complex operations. In this paper, application of MAS for the control of a Microgrid is presented. The MAS approach was selected as a tool, not only to provide intelligence for the needs of complex tasks, but also to facilitate management in the design of the algorithm. In this paper, emphasis is placed on the internal operation of the Microgrid and its participation in the Energy Market. The main idea of the algorithm presented, is that every DER or controllable load decides what is best for it, taking into account the overall benefit through the auction algorithm. Of course, a MAS does not aim exclusively to market participation, but also needs other functionalities. Thus, the proposed architecture is considered as the first step of a more integrated control mechanism. Practical aspects from the implementation of the method on a small laboratory Microgrid and projection to more realistic LV grids are discussed. General conclusions regarding the benefits from this decentralized control approach compared to centralized control are finally presented.

References

[1] EU Research Project Microgrids [Online]. Available: http://microgrids.power.ece.ntua.gr

[2] Microgrids: reliable power in a small package [Online]. Available: http://www.lbl.gov/Science-Articles/Archive/EETD-microgrids.html

[3] R. Lasseter, A. Akhil, C. Marnay, J. Stephens, J. Dagle, R. Guttromson, A. Meliopoulos, R. Yinger, and J. Eto, "White Paper on Integration of Distributed Energy Resources. The CERTS MicroGrid Concept," Consortium for Electric Reliability Technology Solutions (CERTS), CA, Tech. Rep. LBNL-50 829, 2002.

[4] A. Dimeas and N. D. Hatziagyriou, "A Multiagent system for Microgrids," in *Proc. IEEE PES General Meeting*, Denver, CO, Jun. 2004, PESGM2004–001 244.

[5] Z. Ming, R. Jianwen, L. Gengyin, and X. Xianghai, "A multi-agent based dispatching operation instructing system in electric power systems," in *Proc. Power Engineering Society General Meeting 2003*, vol. 1, Jul. 13–17, 2003, pp. 436–440.

[6] J. A. Hossack, J. Menal, S. D. McArthur, and J. R. McDonald, "A multiagent architecture for protection engineering diagnostic assistance," in *Proc. Power Engineering Society General Meeting*, vol. 18, May 2003, pp. 639–647.

[7] T. Nagata and H. Sasaki, "A multi-agent approach to power system restoration," in *Proc. ISAP 2003*, Lemnos, Greece, Sep. 2003.

[8] T. Hiyama, M. Kouno, H. Ono, and K. Furukawa, "Multi-agent based wide area stabilization control of electric power systems," in *Proc. ISAP 2003*, Lemnos, Greece.

[9] J. Ferber, *Multi-Agent Systems. An Introduction to Distributed Intelligence*. Reading, MA: Addison-Wesley, 1999.

[10] J. M. Bradshaw, *Software Agents*. Cambridge, MA: MIT Press, 1997.

[11] D. P. Bertsekas and D. A. Castanon, "A forward/reverse auction algorithm for asymmetric assignments problems," *Comput. Optimiz. and Applic.*, vol. 1, pp. 277–297, 1992.

[12] D. P. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Comput. Optimiz. and Applic.*, vol. 1, pp. 7–66, 1992.

[13] Foundation of Intelligent Physical Agents (FIPA) [Online]. Available: http://www.fipa.org

[14] Java Agent DEvelopment Framework [Online]. Available: http://jade.tilab.com/

[15] FIPA, FIPA 97 Specification Part 2: Agent Communication Language (1997). [Online]. Available: http://www.fipa.org/

[16] JADE PROGRAMMER'S GUIDE [Online]. Available: http://jade.tilab.com/doc/index.html

[17] S. Papathanassiou, D. Georgakis, N. Hatziargyriou, A. Engler, and C. Hardt, "Operation of a prototype microgrid system based on micro-sources equipped with fast-acting power electronics interfaces," in *Proc. 31th PESC 2004*, Aachen, Germany, Jun. 2004.

[18] S. Papathanassiou, N. Hatziargyriou, and K. Strunz, "A benchmark LV microgrid for steady state and transient analysis," in *Proc. CIGRE Symp. Power Systems with Dispersed Generation* , Athens, Greece, Apr. 2005.

[19] A. Tsikalakis and N. D. Hatziagyriou, "Economic scheduling functions of a microgrid participating in energy markets," in *Proc. CIGRE Symp. Power Systems with Dispersed Generation*, Athens, Greece, Apr. 2005.

[20] F. Ygge and H. Akkerman, "Decentralized markets versus central control: A comparative study," *J. Artif. Intell. Res.*, vol. 11, pp. 301–333, Oct. 1999.

Aris L. Dimeas (S'04) was born in Athens, Greece, in 1977. He received the diploma in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece. He is currently pursuing the Ph.D. degree in the Electrical and Computer Engineering Department of NTUA.

His research interests include dispersed generation, artificial intelligence techniques in power systems, and computer applications in liberalized energy markets.

Mr. Dimeas is member of the Technical Chamber of Greece and student member of IEEE.

Nikos D. Hatziargyriou (S'80–M'82–SM'90) was born in Athens, Greece. He received the diploma in electrical and mechanical engineering from the National Technical University of Athens (NTUA), Athens, Greece, and the M.Sc. and Ph.D. degrees from UMIST, Manchester, U.K.

He is currently a Professor in the Power Division of the Electrical and Computer Engineering Department of NTUA. His research interests include dispersed generation, artificial intelligence techniques in power systems, modeling and digital techniques for power system analysis and control.

Dr. Hatziargyriou is a member of CIGRE SCC6 and of the Technical Chamber of Greece.