Renewable Energy 39 (2012) 166–174

Contents lists available at SciVerse ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Securing critical loads in a PV-based microgrid with a multi-agent system

Manisa Pipattanasomporn*, Hassan Feroze, Saifur Rahman

Virginia Tech – Advanced Research Institute, 900 N Glebe Rd, Arlington, VA 22203, USA

ARTICLE INFO

Article history:
Received 21 April 2011
Accepted 28 July 2011
Available online 3 September 2011

Keywords:
Solar photovoltaics
Microgrid
Multi-agent system
Load management
Islanding and resynchronization

ABSTRACT

The objective of this paper is to describe the development of a multi-agent system for the control of a PV-based microgrid. A case study is presented to demonstrate the agents' abilities to island the PV-based microgrid in the event of an external fault, secure critical loads, and resynchronize the microgrid to the main grid after the fault is cleared. Simulation results indicate that the multi-agent system can isolate and stabilize the PV-based microgrid within half an electrical cycle. During this time interval all non-critical loads are shed according to their priorities while all critical electrical loads are secured. This analysis can serve as a guide for the practical implementation of an agent-based approach for resilient operation of a microgrid that has a solar photovoltaic (PV) system coupled with battery storage.

1. Introduction

A multi-agent system is a combination of several agents working together to accomplish a set of assigned tasks [1,2]. In the context of power systems, the multi-agent technology have been applied in a variety of applications, such as to perform power system disturbance diagnosis, fault diagnosis, power system restoration, and system visualization. Multi-agent systems for power engineering applications are discussed in [3,4]. In the context of microgrids, multi-agent systems were implemented to perform energy management of distributed power sources including PV and battery storage [5,6] and energy resource scheduling [7]. In addition, multi-agent systems were implemented to determine optimal microgrid operation [8–11] and restore a distribution system network after a fault [12,13].

In a given situation, an agent must be able to issue a control signal in response to an event sensed from the external environment quickly enough to manage the microgrid in a timely fashion. This paper presents the design and implementation of a multi-agent based controller for controlling a microgrid that contains a solar photovoltaic system coupled with battery storage. In addition to the design of the agent-based system, this paper also evaluates theoretically the time it takes from when an agent issues a control signal to control a physical device in a microgrid to when the command is executed. This study of dynamic interactions – especially during the islanded operation – between the agents and the physical system has not been fully addressed in the previous work reported in the literature. It is expected that this paper can provide useful guidance that can serve as the basis for the practical implementation of an agent-based approach in a microgrid that contains a solar photovoltaic system with energy storage units.

This paper is organized as follows. Section 2 defines a microgrid and sets the stage for the discussion to follow. Section 3 presents agent architecture and specifications, agent collaborative diagram and knowledge modeling. Section 4 describes the case study set up and interactions among agents during an emergency situation to island the PV-based microgrid, secure critical loads during emergency and resynchronize the microgrid when the external fault is cleared. Section 5 presents simulation results and discussions.

2. Overview of a microgrid

In general, a microgrid is a subset of an electric power system. At a minimum, a microgrid comprises a single-customer with internal generation and loads. This is generally known as a single-customer microgrid. A microgrid can comprise a part of the distribution feeder, which is generally known as a partial feeder microgrid; or the whole distribution feeder, known as a full-feeder microgrid. These possible microgrid boundaries are illustrated in Fig. 1.

A microgrid usually has two operating strategies, which include "normal-mode" and "outage-mode" operations. During the normal-mode operation, a microgrid runs as a part of the local utility and coordinates its internal distributed energy resources (DERs) and loads for the most optimal operation. In the case of the

* Corresponding author. Tel.: +1 571 858 3302.
E-mail addresses: mpipatta@vt.edu (M. Pipattanasomporn), hassan.feroze@gmail.com (H. Feroze), srahman@vt.edu (S. Rahman).

0960-1481/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.renene.2011.07.049

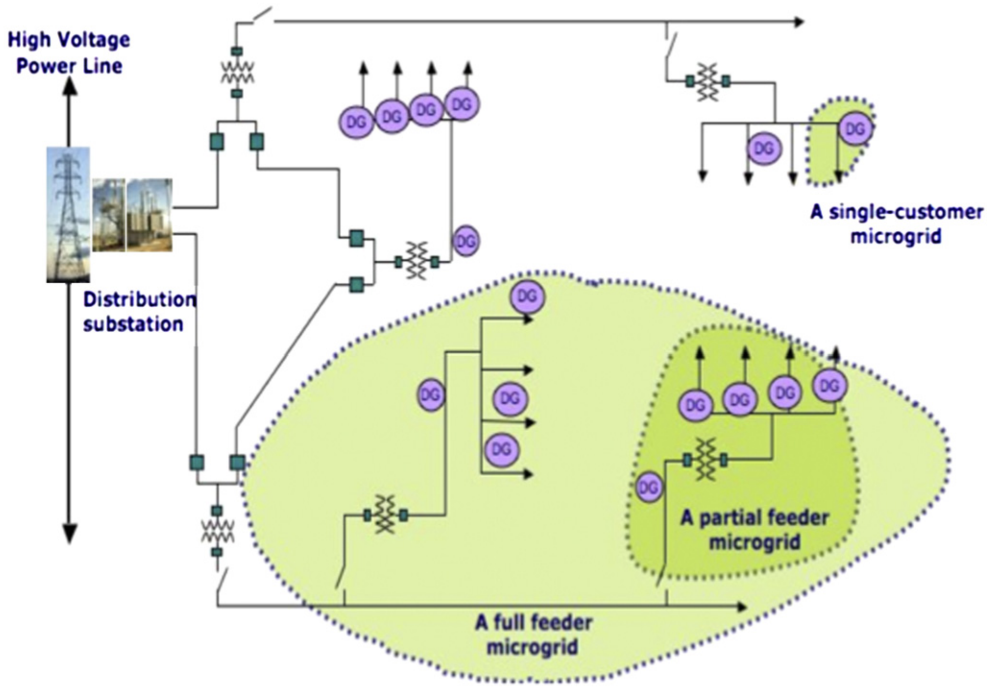

Fig. 1. Microgrid illustration – a single-customer microgrid, a partial feeder microgrid, and a full-feeder microgrid.

outage-mode operation, a microgrid can be designed to isolate itself from the main grid and operate autonomously to secure critical loads. In such a situation, a microgrid controls loads based on a pre-defined prioritized list and activates its internal generation to secure critical loads as needed. A working microgrid should be "intelligent", "distributed" and "autonomous", which hereafter will be referred to as an intelligent distributed autonomous power system (an IDAPS microgrid). Devices in an IDAPS microgrid – namely generators, loads, switches, etc. can communicate among each other through an IP-based network (IP = Internet Protocol) via a multi-agent system. This work focuses on the microgrid operation during emergencies.

3. Design of the multi-agent system

3.1. Agent architecture and specifications

The proposed multi-agent consists of four agents, namely control agent, DER agent, user agent and database agent. The four agents together comprise the multi-agent system. They performs synchronized actions to achieve their goal, which is to island the microgrid and secure critical loads during external fault conditions, and to resynchronize the microgrid to the main grid after the fault is cleared. The architecture of the proposed IDAPS multi-agent system is presented in Fig. 2.

The 'control agent' is responsible for monitoring the health of the utility network. Once contingency situations or grid failures are detected, it sends signals to isolate the microgrid from the utility. It also detects the restoration of the upstream grid, issues the resynchronization signal, and informs all other agents of the main grid's restoration status.

The 'DER agent' is responsible for monitoring and controlling DER power levels and its connect/disconnect status. DER information to be stored may include DER identification number, type (solar cells, microturbines, fuel cells, etc), power rating (kW), local fuel availability, cost function or price at which users agree to sell, as well as DER availability, i.e. planned maintenance schedule. In this paper, DER in consideration is a solar PV unit with battery storage.

The 'user agent' acts as a customer gateway that makes features of an IDAPS microgrid accessible to users. This agent monitors voltage, current, active and reactive power consumption at each critical and non-critical load. A user agent also allows users to control loads based on the priority pre-defined by a user.

The 'database agent' is responsible for storing system information, as well as recording the messages and data shared among agents. This agent also serves as a data access point for other agents, and keeps track of all available agents and their capabilities.

The proposed multi-agent system is designed to follow the IEEE's standard on Foundation for Intelligent Physical Agents (FIPA).

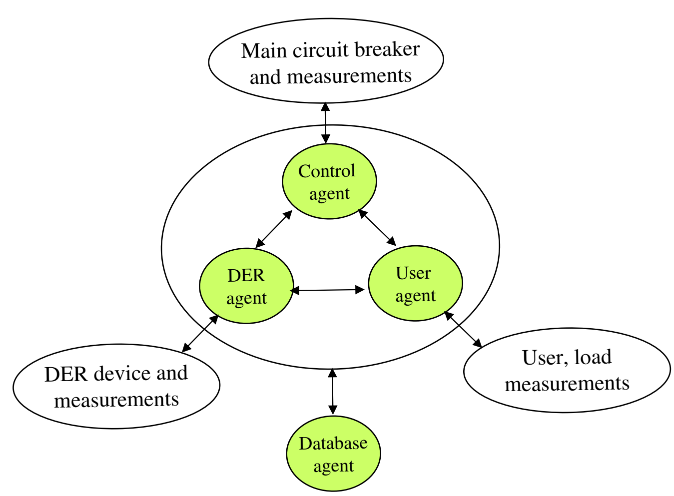

Fig. 2. IDAPS agent architecture and their interaction, where arrows represent messaging exchange among agents via Transmission Control Protocol/Internet Protocol (TCP/IP).

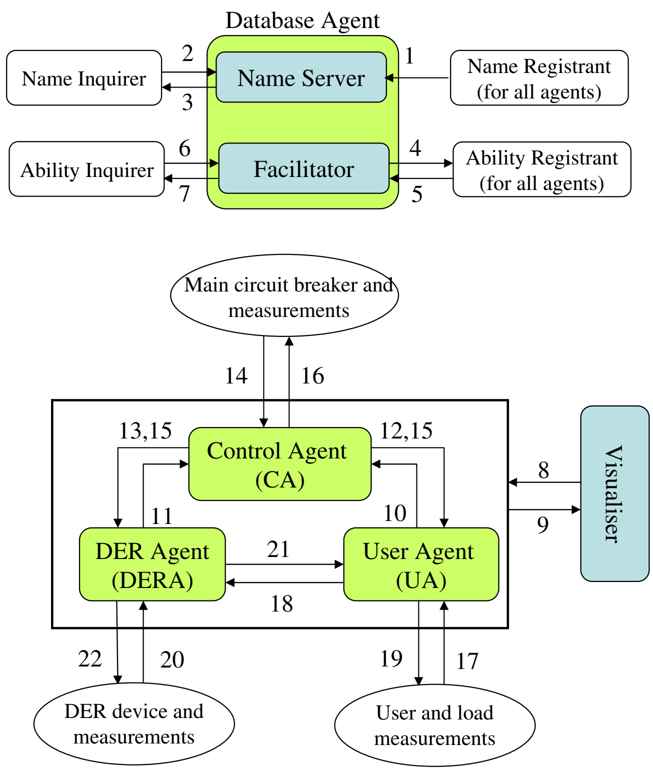

Fig. 3. Agent's collaborative diagram – Numbers 1–22 represent interactions among agents or between the agents and their environment. See Table 1.

This is to ensure interoperability among different systems and platforms so that the proposed multi-agent system can be universally accepted.

3.2. Agent collaborative diagram

The collaborative diagram defines the interaction among agents. It also defines the interaction between an agent and its environment, i.e. how the control agent interacts with the distribution circuit and senses circuit information; how the DER agent interacts with DER devices and obtains DER information; and how the user agent interacts with users and measures power consumption at the loads. Fig. 3 illustrates the IDAPS agent's collaborative diagram.

The diagram illustrates all four IDAPS agents – a control agent, a DER agent, a user agent, as well as a database agent. The database agent comprises the name server and the facilitator. The name server is responsible for allowing all agents in the system to register and maintaining directory of the agents and their locations. The facilitator contains information regarding abilities of all agents in the multi-agent system. The numbers (1–22) in Fig. 3 represent the interactions among agents in the IDAPS microgrid, and the interactions between the agents and their environment. These interactions are summarized in Table 1.

In short, the IDAPS agents start the process by initializing the system. The initialization of the IDAPS multi-agent system is performed by all agents notifying the database agent of their presence (1–7). The visualizer receives copies of all messages exchanged within the multi-agent system and is responsible for displaying these messages to all agents in the system (8–9). After the initialization process, the user agent and the DER agent register with the control agent (10–13), and perform internal communications among agents (15, 18, 21), as well as external communications with their environments (14, 16, 17, 19–22). This discussion refers to the

Table 1
IDAPS Agents' Interactions in the Collaborative Diagram.

	Collaboration	Explanation
1	Registration	Agents notify the database agent of their presence
2	Location and type query	Request for the location of agents and types
3	Location and type response	Response from agents for their locations and types
4	Ability request	Request for agents' abilities
5	Ability response	Response from agents about their abilities
7	Find request	Request to obtain a list of agents with particular abilities
6	Find response	Response containing a list of agents matching the desired criteria
8	Activity request	Request for all agents' activities
9	Activity notification	Response containing all messaging exchanged among agents or between agents and their environment
10	UA registration request	Request from a user agent, containing its ID, to the control agent for subscription
11	DERA registration request	Request from a DER agent, containing its ID, to the control agent for subscription
12	CA registration response	Acknowledgement from the control agent that the user agent's ID has been stored.
13	CA registration response	Acknowledgement from the control agent that the DER agent's ID has been stored.
14	CA information retrieval	Retrieval of information from the main grid by the control agent
15	CA inform signals	Inform signals issued by the control agent to inform all registered subscribers about the information retrieved from the main grid
16	CA control signals	Control signals issued by the control agent to open/close the main circuit breaker
17	UA information retrieval	Retrieval of the loads' power requirement from the external circuit by the user agent
18	UA load information	Information sent by the user agent to inform the DER agent the loads' power requirement
19	UA control signals	Control signals issued by the user agent to turn the load ON or OFF
20	DERA information retrieval	Retrieval of information from DERs by the DER agent
21	DER output information	Information sent by the DER agent to the user agent about the amount of power produced
22	DERA control signal	Control signals issued by the DER agent to control power production from DER

agents' interactions in a single IDAPS multi-agent system. More than one multi-agent system can be created using the same collaborative diagram.

3.3. Knowledge modeling

In knowledge modeling, "Facts" are defined for each agent's application. Facts represent statements that an agent believes to be true, either about itself or its external environment. These Facts are collectively called ontology. Ontology is the vocabulary used by agents during communication. Ontology is created for the multi-agent system based on the application design process mentioned in [13]. The application consists of both abstract and physical concepts. For instance, the island mode information is abstract, while energy is a physical concept. Table 2 lists Facts defined in the proposed multi-agent system, together with their attributes and default values.

4. Description of the cyber-physical systems and their interactions

The system under study comprises both physical and cyber (virtual) systems. The physical system is the microgrid – developed in the Matlab/Simulink environment. The cyber system is the multi-agent system – developed in an external programming language. The connections between the cyber and physical systems

Table 2
Facts in the IDAPS Multi-Agent System.

Facts	Attributes	Default value
islandmode	is_island: Boolean	is_island: false
	main_cb_status: Boolean	main_cb_status: true
	AgentsID: String	AgentsID: “ ”
	outageDuration: String	outageDuration: “ ”
	outageHour: String	outageHour: “ ”
criticalCB	cl_cb_status: Boolean	cl_cb_status: true
noncriticalCB	ncl_cb_status: Boolean	ncl_cb_status: true
DERspecs	sendDERinfo: Boolean	sendDERinfo: false
	DER_ID: String	DER_ID: “ ”
	DER_type: String	DER_type: “ ”
	DER_rating: String	DER_rating: “ ”
	DER_fuel_avai: String	DER_fuel_avai: “ ”
	DER_Cost: String	DER_Cost: “ ”
	UserAgentID: String	UserAgentID: “ ”
ua_DERcmd	ua_required_power: String	der_require_power: 0.0
	sendDERcmd: Boolean	sendDERcmd: false
	UserAgentID: String	userAgentID: “ ”
re_uaDERcmd	ack_required_power: String	require_power: 0.0
	UserAgentID: String	userAgentID: “ ”
dg_DERcmd	der_produced_pwr: String	der_produced_pwr: 0.0
	send_der_cmd: Boolean	send_der_cmd: false
	DERAgentID: String	DERAgentID: “ ”
re_dgDERcmd	ack_produced_pwr: String	produced_pwr: 0.0
	DERAgentID: String	DERAgentID: String
agentsName	AgentID: String	AgentID: “ ”
agentRegistered	AgentID: String	AgentID: “ ”
	isRegistered: Boolean	isRegistered: false
inIDAPS	AgentID: String	AgentID: “ ”
	isInIDAPS: Boolean	isInIDAPS: false

are established via TCP/IP connections. The case study was set up such that the physical system is located in one computer and the cyber system is located in the other. This section describes how the physical and cyber systems are set up and interconnected.

4.1. Physical system

4.1.1. Circuit description

To demonstrate the agent's ability to perform microgrid control and management, a residential community microgrid served by a 25 kVA distribution transformer is used in this case study. The community comprises a group of homes with multiple types of loads, as well as some internal generation sources or distributed energy resources. See Fig. 4.

4.1.2. Distributed energy resources (DER)

In this simulation, the DER is solar panels coupled with battery storage. Their sizes are chosen to ensure sufficient energy and capacity to secure all critical electrical loads during an outage. The issue of determining the proper PV and battery sizes is not the focus of this paper, and will not be discussed here. The model of a PV generator is developed in Matlab/Simulink. It is designed based on the mathematical relationships presented in [14,15]. The generic battery model available in Matlab/Simulink [16] is used in the simulation. During the grid-connected mode, the grid interface unit controls the voltage magnitude, frequency and phase of the DC DER to follow those of the main grid at all times. During the islanded mode, the grid interface unit keeps the DER's voltage magnitude at 1 per unit and its frequency at 60Hz. The grid interface unit used in this study is developed according to the methodology presented [17].

4.1.3. Load profile and their priority

The actual hourly load data of a 25kVA distribution transformer obtained from a local electric utility [UTIL] are used as a basis for the simulation. Per UTIL data, on average, a 25kVA distribution transformer serves a group of three homes. According to the EPRI's RELOAD database [18], residential loads are classified into nine types. These are refrigeration, freezer, cooking, lighting, space cooling, space heating, water heating, clothes drying and others. The percentage breakdown of the total consumption by load types is available from the RELOAD database. The community load profile (in August) by load types is then derived using the actual UTIL's load profile and the RELOAD's percentage breakdown of electrical power consumption by load type, as shown in Fig. 5.

In 2007, Pacific Northwest National Laboratory [19] conducted a survey that included roughly 250 customers. The survey indicated types of household appliances for which customers are willing to change their usage behaviors in response to variable electricity rates. Based partly on the survey results, this study considers the load priorities as follows:

- Critical loads: refrigeration, freezer, cooking, and lighting
- Non-critical loads: space cooling, water heating, dryer and others

In this case, the peak demand for critical loads served by this transformer is 6.2 kW. The total load is 14.5 kW. This study assumes that the devices to disconnect non-critical loads are readily available.

4.2. Cyber (virtual) system

4.2.1. The IDAPS multi-agent system

In addition to the physical system, Fig. 4 also illustrates the IDAPS multi-agent system – represented by a laptop computer. These agents perform the following actions: island the microgrid once an external fault is detected; secure critical loads during islanded operation with available electricity from the PV and energy storage unit; and resynchronize the microgrid to the utility network once the external fault is cleared.

4.2.2. Agent interactions once an external fault is detected

Agents are currently designed to island the microgrid upon detection of an external fault. Once the system voltage goes below a certain value, the agent issues control signals to disconnect the main circuit breaker, and balance the demand with the available supply (in this case from a solar photovoltaic system and battery storage) by disconnecting non-critical loads. Some critical loads can be disconnected based on their pre-set priority if necessary.

Once an external fault is detected, the agent interactions can be summarized into the following four sequential actions. At the occurrence of any of these steps, the “Facts” (as shown in Table 1 in the multi-agent ontology are updated. The updated ontology helps agents to understand what happens in their environment. During the simulation, the time for message exchanges among agents to accomplish any given task is less than a tenth of a second.

- *Action 1: The control agent monitors and detects external faults.* This can be accomplished by sensing the system voltage in a simulated microgrid. The fault condition is detected when the voltage level goes below a pre-set threshold.
- *Action 2: The control agent issues the status change signal and opens the main circuit breaker.* Once the external outage is detected, the control agent sends a signal to notify all agents in the network of the microgrid status and the time of outage. The message consists of the updated '*islandmode*' Fact, of which the '*is_island*' attribute is set to 'true'; the '*outageHour*' attribute is set to the time when fault was detected. At the same time, the control agent also opens the main circuit breaker to isolate the

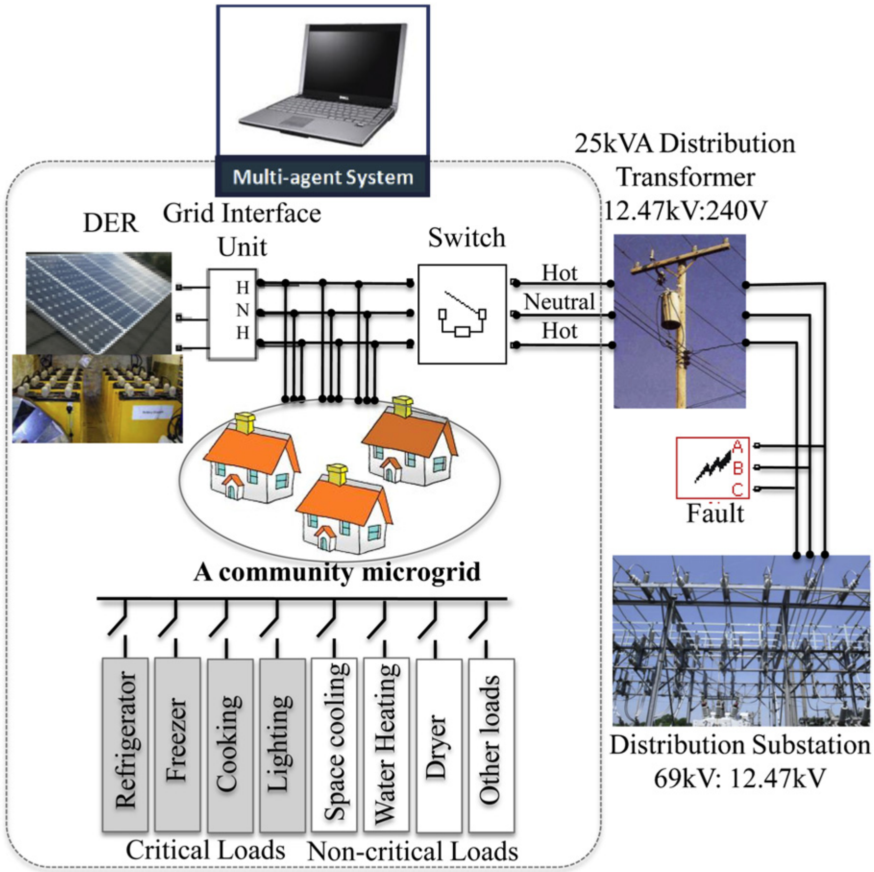

Fig. 4. A simulated microgrid.

microgrid from the main grid. The *'main_cb_status'* is changed to 'false'.

- *Action 3: User agents secure critical loads and shed non-critical loads at the instant of the outage.* Once an outage is detected, all user agents disconnect all non-critical loads. This is accomplished by updating the *'noncriticalCB'* Fact to 'false'.
- *Action 4: User agents communicate with the DER agent to balance the demand with the available supply during an outage.* During

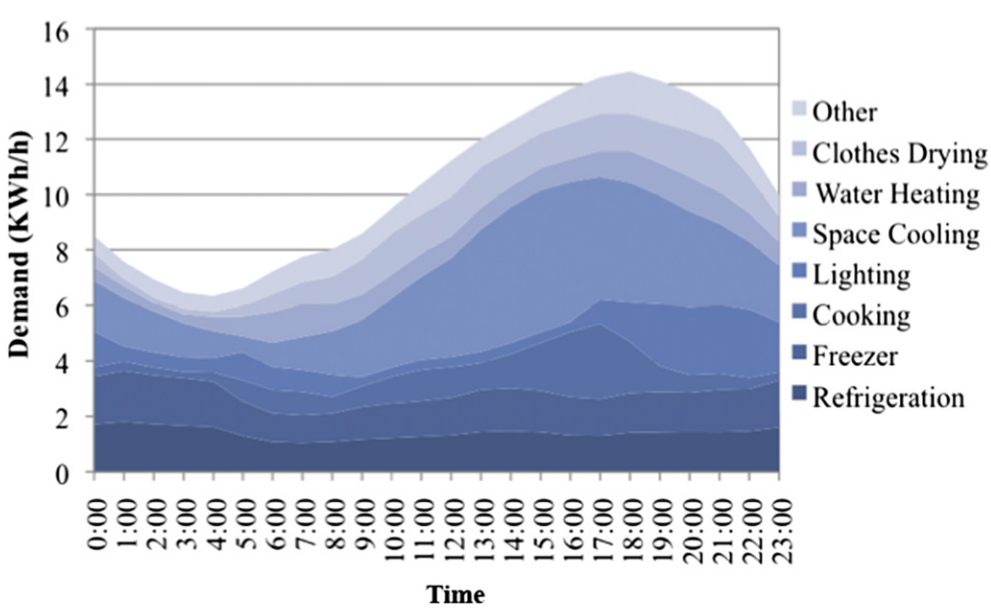

Fig. 5. Load profile by load type of a community served by a 25 kVA transformer.

Table 3
Facts Updated During Emergencies.

Facts	Updated attributes	Updated value
islandmode	is_island	True
	main_cb_status	False
criticalCB	cl_cb_status	True
noncriticalCB	ncl_cb_status	False
ua_DERcmd	ua_required_power	is set equal to the power required by the critical loads
	UserAgentID	is set equal to the user agent's ID
re_uaDERcmd	ack_required_power	is set equal to the information received
	UserAgentID	is set equal to the user agent's ID
dg_DERcmd	der_produced_pwr	is set equal to the power required by the critical loads of the related user agent
	DERAgentID	is set equal to the DER agent's ID
re_dgDERcmd	ack_produced_pwr	is set equal to the information received
	DERAgentID	is set equal to the DER agent's ID

Table 4
Facts Updated During the Resynchronization.

Facts	Updated attributes	Updated value
islandmode	is_island	False
	outageHour	N/A
criticalCB	cl_cb_status	True
noncriticalCB	ncl_cb_status	True
ua_DERcmd	ua_required_power	is set to shut down or reduce, depending on the preference
	UserAgentID	is set equal to user agent's ID
re_uaDERcmd	ack_required_power	is set equal to the information received
	UserAgentID	is set equal to user agent's ID
dg_DERcmd	der_produced_pwr	is set to reduce the output
	DERAgentID	is set equal to the DER agent's ID
re_dgDERcmd	ack_produced_pwr	is set equal to the information received
	DERAgentID	is set equal to the DER agent's ID

an emergency when the electricity supply from the grid is not available, the user agents constantly communicate and exchange messages with the DER agent, of which the purpose is to stabilize the microgrid by balancing the demand with the available supply from the locally available PV and storage energy sources. This is accomplished by the user agents who send the updated *'ua_DERcmd'* to the DER agent to inform the agent of their current load status. Upon receiving the demand signals from user agents, the DER agent acknowledges the demand signal by updating the 're_*uaDERcmd'*. At the same time, the DER agent then controls and measures the DER's power and energy output availability and reports the DER's status by sending the *'dg_DERcmd'* to all user agents. User agents acknowledge this signal by updating the 're_*dgDERcmd'*.

At the occurrence of any of these steps, the Facts (as shown in Table 3) in the multi-agent system's ontology are updated. The updated ontology helps agents to understand what happens in their environment. The updated Facts during this process are summarized in Table 3.

4.2.3. Agents' interactions once the external fault is cleared

The multi-agent system also plays a key role in reconnecting a microgrid to the main grid after the external fault is cleared. During the islanded operation, the agent monitors the voltage, frequency and phase angle at both sides of the interconnection point (across the switch as shown in Fig. 4). For the agent to issue the resynchronization signal, the voltage and frequency of the microgrid must be as close to those of the main grid as possible. Having confirmed that the voltage and frequency differences across the switch are smaller than the pre-set thresholds, the control agent issues control signals to synchronize the microgrid to the utility. In this case, the '*is_island*' attribute of the '*islandmode*' Fact is set to 'false' (to synchronize the two systems), and the'*criticalCB*' and *'noncriticalCB'* Facts are set to 'true' (to reconnect the non-critical loads). The DER agent talks to the user agents to shut down or reduce the DER power output. This can be accomplished by agents exchanging messages to update the *'dg_DERcmd'* and *'ua_DERcmd'* Facts. The updated Facts during this process are summarized in Table 4.

4.3. Connection between the cyber and the physical systems

The multi-agent system is connected to the microgrid in the MATLAB/Simulink environment over a TCP/IP connection. Fig. 6 depicts this scenario.

A third party TCP/IP server [20] implementable in MATLAB Simulink is used to establish the TCP connectivity. Socket programming is carried out in agents' external java classes. The third party TCP/IP server is capable of managing multiple outputs and unlimited number of string inputs. However, the server allows only a single TCP client connection at a time and follows a specific format for the input/output messages. This limits our control over the connection between the multi-agent system and the microgrid because each agent (a user agent, a DER agent and a control agent) requires a separate TCP connection to the microgrid. To handle the situation a middle server is developed.

The middle server allows multiple TCP connections to the microgrid and increases flexibility to control the microgrid with the designed multi-agent system. Any messages sent from the multi-agent system are processed in the middle server and sent out to the microgrid's TCP/IP server in an appropriate format. Similarly, messages sent by the microgrid's TCP/IP server are received by the middle server and sent out to the appropriate receivers in the multi-agent system.

5. Results and discussion

5.1. Scenario description

The circuit shown in Fig. 4 is simulated for 24 h, assuming that the outage occurred at 5 pm for 3 h. The objective of this case study is to analyze the functionality of the proposed multi-agent system to: island the microgrid once a fault is detected; secure critical loads during an emergency, and resynchronize the microgrid to the main grid after the external fault is cleared. The assumption is that there is no limitation on energy availability to serve critical loads during the entire outage duration.

5.2. Simulation results and discussion

The simulation result is illustrated in Fig. 7. As soon as the outage is detected at 5 pm, the multi-agent system performs its actions to shed all non-critical loads and secure critical loads according to the pre-defined prioritized list.

In this case, the critical loads served are refrigerator, freezer, cooking and lighting; and all non-critical loads are shed.

During the Grid-Connected Mode (before 5 pm): The microgrid's voltage and frequency are controlled such that they follow the grid's voltage and frequency, which are roughly at 1 per unit and 60 Hz, respectively. Fig. 8 presents a snapshot at the instant of the outage at around 5 pm, which illustrates the fluctuation in voltage and current at the interconnection point between 0.1 s before the fault and 0.2 s after the fault. As shown, the voltage is always kept at around 1 per unit. The current indicates the current produced by the DER. Before the external fault occurs, the DER unit supplies

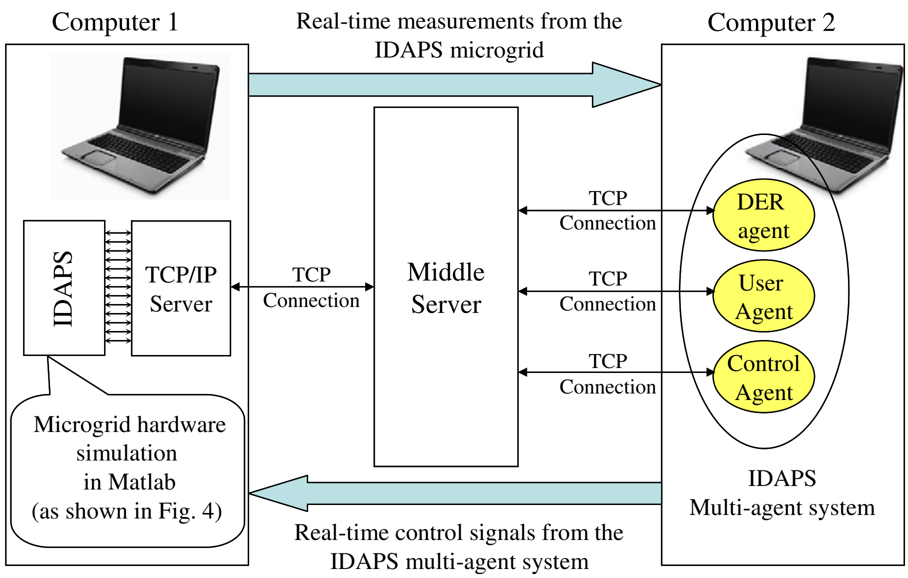

Fig. 6. Middle server implementation.

some power to the microgrid. This is indicated by the non-zero current injecting into the grid at the interconnection point.

During the Transition (at 5 pm): Once the control agent detects the fault at 5 pm, the control agent informs the user agents and the DER agent. The user and DER agents exchange information and determine the amount of loads to be shed and the amount of power and energy required internally in order to stabilize the microgrid during the outage. At the same time as soon as the external outage is detected, the control agent sends a control signal to isolate the microgrid by disconnecting the main circuit breaker. Depending on the pre-defined load priority and the available internal electricity supply, the user agents can disconnect non-critical loads and certain critical loads. In this case, it is assumed that the DER can fully support connected critical loads without any energy limitation during the outage. It is also assumed that the DER can increase its output instantaneously with the help of battery storage once the microgrid is islanded. Thus, the user agents only disconnect the non-critical loads and leave the critical loads intact. All agent actions – from detecting the fault, disconnecting the main circuit breaker, disconnecting the non-critical loads to stabilizing the grid – can be accomplished within about half an electrical cycle, i.e. less than 0.008 s for a 60-Hz system. This finding is as a result of multiple simulation runs. The DER fault current as shown is limited by the inverter rating in this case of this PV-based microgrid.

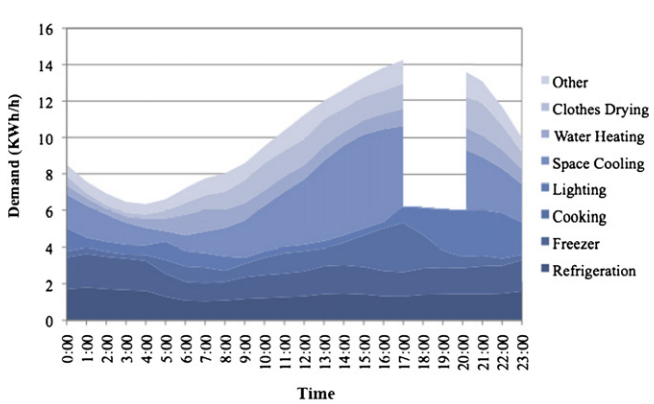

Fig. 7. Load profile with agents securing critical loads during the outage between 5 and 8pm. All non-critical loads are shed.

During the Islanded Mode (5 pm – 8 pm): After the fault occurred at 5 pm, the IDAPS microgrid is disconnected from the main grid. When the microgrid is operated in the islanded mode, the grid interface unit controls the DER output voltage at 1 per unit and the frequency at 60Hz. The user agents and the DER agent balance the demand and supply within the microgrid. The total system load is reduced to its critical loads and the DER produces almost at its full capacity internally to supply these critical loads. See Fig. 9 – a snapshot at the instant of the restoration of the utility grid at around 8 pm, which illustrates the fluctuation in voltage and current at the interconnection point between 0.1 s before and 0.2 s after the resynchronization.

The first graph in Fig. 9 illustrates the fluctuation in voltage and current at the interconnection point in per unit. The second graph in Fig. 9 displays the DER output power (in per unit) and the DER agent command to control DER output. The third graph in Fig. 9 measures the voltage difference between the RMS voltage at the microgrid side and the RMS voltage at the main grid side in volts. During the supply-side outage (5 pm – 8 pm), the main grid voltage collapses, while the microgrid maintains its voltage at 1 per unit. Therefore, there are some voltage differences between the main grid and the microgrid, as shown in Fig. 9, before 8 pm. This is when the microgrid is operated in the islanded mode.

During Resynchronization to the Main Grid (at 8 pm): As the external circuit is restored at 8 pm, the main grid voltage gradually increases. Therefore, the voltage difference, as shown in Fig. 9, gradually decreases at around 8 pm. The control agent monitors this voltage difference, as well as the frequency and phase angle of both systems. Once it sees the voltage and frequency differences drop below certain thresholds, the control agent issues control signals to: (i) close the main breaker; (ii) reconnect non-critical load circuit breakers; and (iii) decrease DER output to prevent instantaneous overshoot in the system's voltage and current.

In this case, at 0.02 s after 8 pm, the control agent issues the synchronization signal. This implies the agent waits 0.02 s to ensure that the voltage and frequency of both systems are similar. After the control agent issues commands to resynchronize the two systems, it can be seen that the system voltage is stabilized at 1 per unit. There is minor current overshooting in the first half a cycle. See the

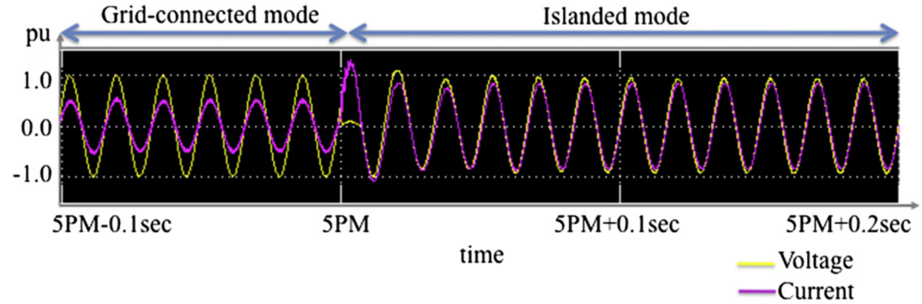

Fig. 8. The variation of 60 Hz voltage and current waveforms measured at the interconnection point before and after the upstream outage applied to the circuit.

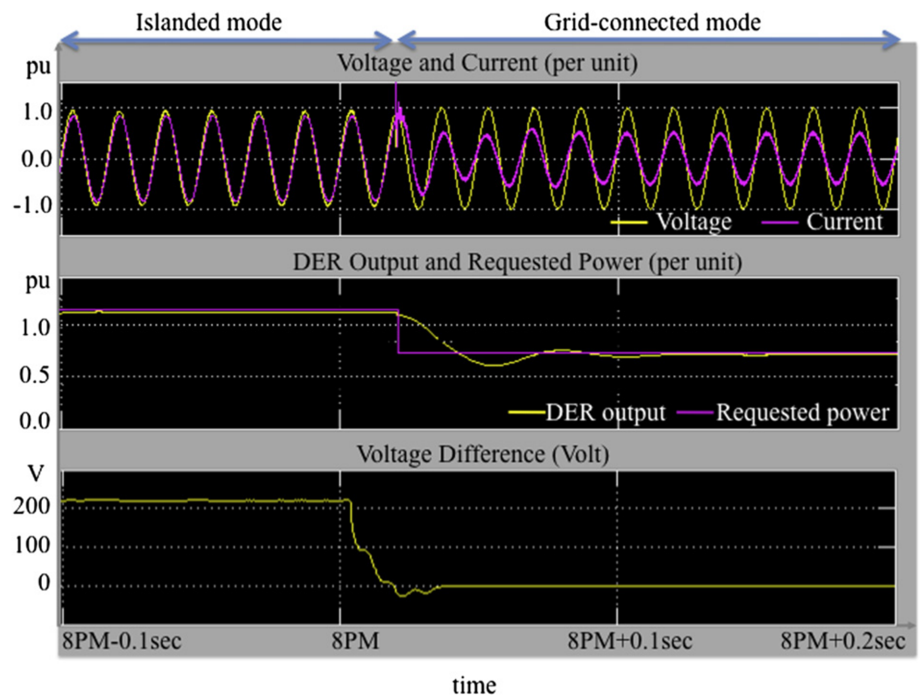

Fig. 9. Simulation results during the synchronization of the microgrid to the main grid. This shows a series of event before and after the utility voltage is restored.

first graph in Fig. 9. The DER output demonstrates a small ripple for about 0.1 s before arriving at its steady state within 0.2 s later. See the second graph in Fig. 9. The DER current as shown after 8 pm implies that the DER resumes to supply some part of the load under normal operating conditions.

It should be emphasized that, although the message delivery time among agents is not of concern in the presented simulated environment, this factor must be considered in an actual system implementation to ensure satisfactory real-time performance.

6. Conclusions

A multi-agent system was designed and implemented in a simulated environment to control and manage a PV-based microgrid during an emergency condition. The system comprises both physical and cyber systems. The physical system is the model of a community microgrid with a solar PV system and energy storage. The cyber system is the multi-agent system developed using an external programming language.

This paper demonstrates the agent's abilities to isolate the PV-based microgrid and secure critical loads during fault conditions, as well as resynchronizing the microgrid after the external fault is cleared. In particular, the case study addresses how the agent can sense the anomaly in the voltage level from the main grid; as well as its ability to exchange messages in order to achieve the agents' common goal in a timely manner. Research findings indicate that, theoretically, it takes – on average over multiple simulation runs – about half an electrical cycle to stabilize the microgrid during both the islanding and the resynchronization. This paper is based on the operation of a microgrid with PV and battery storage. Using a different type of DER technology in a microgrid may result in different responses especially during the transitions.

Funding source

This work is supported in part by the U.S. National Science Foundation under grant ECCS-0901410.

References

[1] Nwana HS. Software agents: an overview. The Knowledge Engineering Review 1996;11(3):205–44.

[2] Bradshaw JM. Software agents. Cambridge, MA: MIT Press; 1997.

[3] McArthur SDJ, Davidson EM, Catterson VM, Dimeas AL, Hatziargyriou ND, Ponci F, et al. Multi-agent systems for power engineering applications—Part I: concepts, approaches, and technical challenges. IEEE Transaction on Power Systems 2007;22(4):1743–52.

[4] McArthur SDJ, Davidson EM, Catterson VM, Dimeas AL, Hatziargyriou ND, Ponci F, et al. Multi-agent systems for power engineering applications—Part II: technologies, standards, and tools for building multi-agent systems. IEEE Transaction on Power Systems 2007;22(4):1753–9.
[5] Lagorse J, Paire D, Miraoui A. A multi-agent system for energy management of distributed power sources. Renewable Energy 2010;35:174–82.
[6] Jun Z, Junfeng L, Jie W, Ngan HW. A multi-agent solution to energy management in hybrid renewable energy generation system. Renewable Energy 2011;36:1352–63.
[7] Logenthiran T, Srinivasan D, Khambadkone A. Multi-agent system for energy resource scheduling of integrated microgrids in a distribution system. Electric Power Systems Research 2011;81(1):138–48.
[8] Kueck JD, Staunton RH, Labinov SD, Kirby BJ. Microgrid energy management system. ORNL/TM-2002/242. CERTS; 2003.
[9] Phillips L, Link M, Smith R, Weiland L. Agent-based control of distributed infrastructure resources. SAND2005–7937. Sandia National Laboratories; 2006.
[10] Dimeas A, Hatziargyriou ND. Operation of a multiagent system for microgrid control. IEEE Transaction on Power Systems 2005;20(3):1447–55.
[11] Dimeas A, Hatziargyriou ND. Operation of a multiagent system for microgrid control. Power Engineering Society General Meeting 2004;vol. 1: 55–8.
[12] Nagata T, Tao Y, Sasaki H, Fujita H. A multi-agent approach to distribution system restoration. Power Engineering Society General Meeting 2003;vol. 2:660.
[13] Pipattanasomporn M, Feroze H, Rahman S. Multi-agent system: design and implementation. In: Proc. IEEE Power system conference and exposition; 2009. Seattle, WA.
[14] Lorenzo E. Solar electricity engineering of photovoltaic systems. Earthscan Publication Ltd; 1994.
[15] Hansen A, Lars P, Hansen H, Bindner H. Models for a Stand-Alone PV system [Online]. AvailableRetrieved: April 2011. Roskilde: Risø National Laboratory, http://130.226.56.153/rispubl/solenergi/sec-r-12.pdf; 2000.
[16] Matlab/Simulink SimPowerSystems Reference. pp. 2-39–2-56.
[17] Matlab/Simulink SimPowerSystems Demo: Two-level PWM converter and dead time, Ver. 2008b.
[18] RELOAD Database documentation and evaluation and use in NEMS [Online]. Available, http://www.onlocationinc.com/LoadShapesReload2001.pdf; April 2011. Retrieved.
[19] Hammerstrom DJ, Brous J, Chassin DP, Horst GR, Kajfasz R, Michie P, et al. Pacific Northwest Gridwise Testbed Demonstration Projects: Part II. Grid Friendly Appliance Project, PNNL-17079; 2007.
[20] Zimmermann W. IOlib – Hardware Input/Output Library for Matlab and Simulink [Online]. Available, http://www.it.hs-esslingen.de/~zimmerma/software/index_uk.html; April 2011. Retrieved.