



Decentralized control architecture for multi-authoring microgrids

Seyed Hossein Alavi Soltani¹ · Saeed Jalili¹ ·
Mohammad Kazem Sheikh El Eslami¹

Received: 12 March 2023 / Accepted: 4 July 2023 / Published online: 17 July 2023

© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2023

Abstract

A prosumer is a consumer who uses tiny, renewable electricity generation units and, in addition to consumption, can also generate electricity. With the increase in the number of prosumers in a power grid, it is expected that the paradigm of network utilization, at least at the distribution level, would change from centralized to decentralized utilization. In this new paradigm, microgrids are essential in the operation of the whole grid. The decentralized control of a stable network of microgrids (i.e., minimal power outages and fluctuations) is a significant challenge. In this paper, we present an architecture for decentralized control that consists of intelligent agents that manage the distribution network provided by the microgrids at the highest level and houses and their devices at the lowest level. The agents, managed by different private companies, dynamically organize themselves in units called holons, follow their defined policies, and can most tolerate network disruptions. In this architecture, self-adaptive agents will play a key role in sustaining network performance by controlling energy consumption and exchange (i.e., in the event of a shortage in a part of the distribution network). In the end, by simulating the architecture, the capabilities are shown.

Keywords Decentralized control · Holonic architecture · Microgrid · Multi-agent systems · Self-adaptive systems

Mathematics Subject Classification 22-04 · 28-04

✉ Saeed Jalili
sjalili@modares.ac.ir

Seyed Hossein Alavi Soltani
s.alavisoltani@modares.ac.ir

Mohammad Kazem Sheikh El Eslami
aleslam@modares.ac.ir

¹ Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Jalal AleAhmad, Tehran, Tehran 1411713116, Iran

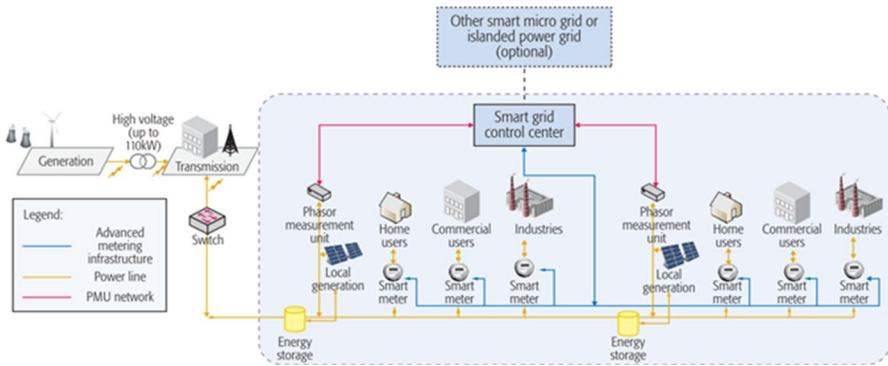


Fig. 1 The microgrid architecture [1]

1 Introduction

Today, due to the significant increase in distributed generation in the power grid and the development of telecommunication and information technologies, the control of power distribution networks is rapidly moving towards pervasive decentralized methods. The microgrid, whose organization is shown in Fig. 1 [1], is an autonomous power grid that has a comprehensive information layer to coordinate, monitor, and control all organized energy sources, so the architecture of the communication network and its main components must be adaptable and flexible [2].

The low-voltage distributed network consists of customers buildings that are connected to at least one shared distributed generation source(DG) and include a variety of distributed energy resources (DERs) such as loads, microresources (MS), and Energy Storage Systems(ESS) [2]. Microgrid works in connection with the main grid, but in case of disconnection from the main grid, it can provide the minimum level of service, and if failure disappears, it could reconnect to the main grid. Each microgrid has central control, and all microgrid devices are connected. It usually performs operations management (such as DER management and control) in a microgrid, controls the energy flow, and connects to the main grid to ensure different DERs have proper operations and communications [2]. Different microgrids may interact with each other on a large geographical scale (such as a city) to form a medium-voltage network. Also, there may be several administrative authorities operating the grid(multiauthority), where each one defines their goals over a grid part (an appliance, a house, or a district of a city) [3]. Centralized grid control for multiauthority (such as private microgrid-owned companies) is not possible or very complex; thus, scalable models for decentralized (distributed) control of large-scale microgrids with a large number of heterogeneous local control centers [3–6] with multiauthority [3, 7, 8] have so far been published that they have attention to time constraints and quality of service. In [3, 5, 9], concerning multi goals in the control system, such as welfare, stability, and security, the new architecture has been presented, and a context-aware ontology for the control subsystem in [1] has been defined. Multi-Agent Systems (MASs) are the most suitable approach for distributed

energy management strategy, as they comprise a group of agents capable of autonomously responding to their environment, perceiving and communicating with it, sensing it, engaging with it, and acting on it [37]. In Sect. 2, a survey of microgrid control systems is presented. In Sect. 3, the proposed architecture, DCAM, is given. We show the process of formation and membership in holon in Sect. 4, the architecture and tasks of control agents presented in Sect. 5, and the designing adapt program presented in Sect. 6. We present in Sect. 7, DCAM simulated results, and in Sect. 8, we evaluate from different dimensions. Finally, we conclude in Sect. 9.

2 Related work

Microgrid control systems operate locally (due to the increase in production of private consumers in the smart grid and the personal goals of the microgrid owners), but to achieve the macro goals of the smart grid, they need to perform their tasks in coordination with each other. As a result, centralized architectures [1, 10], decentralized [5, 6, 9, 11–22] and hierarchical holonic [3, 7, 23–25] for the microgrid control system published and standards for network architecture such as SGAM [26] are presented in both centralized and decentralized forms. In centralized control, there is only one ruler and policymaker for the network. All network operations are performed on the central server(s), so if one or more of the central servers fail or become inaccessible, the smart grid becomes unstable and decreases productivity. In decentralized control, the management and control of each microgrid are done locally by a private company (on that company's server(s)) under that company's policies. Still, global microgrid affairs (such as failure, power shortage management, etc.) are managed by interacting with other microgrids. In the hierarchy control, controllers have a different layer that works together. Each layer has a specific task that particular agents coordinate with the top and down layers' agents. At the lowest layer, managed devices exist. So if the connection with each layer is interrupted, the system may not work well. In holonic architecture, each holon has a fractal structure composed of several other holons, which is part of a higher-level holon. Holons have a recursive and similar structure, and each could have different management. Each holon has the following three characteristics [32]: (1) Stability, (2) Having the capability of autonomy and self-organization to achieve its own goals, and (3) the ability to cooperate with other holons based on common goals. Holonic is similar to hierarchical, but we could have different holons in each hierarchical layer; Each one has just one dominator to manage holon members. Holon dominators also can join its holon to top-layer holons. Tom et al. [28] propose an Agent Negotiation system for demand reduction. This negotiation is done with hundreds and thousands of homes, helping Utilities to meet the supply–demand effectively, thus ignoring the collective potential of coordinated appliance scheduling among multiple households. The goals of householders cannot change on the run time, and if one of the upper-level agents goes down (e.g., Fog Agent), no self-organization scenario is proposed. Fanitabasi et al. [29] presented a novel appliance-level flexible scheduling framework based on consumers' self-determined flexibility and comfort requirements. This framework relies on a decentralized network of autonomous scheduling agents representing a residential

household with multiple appliances. These agents interact and coordinate to select a subset of consumers' schedules and optimize that to reduce demand peaks. The coordinated plans are submitted back to the scheduling agent. Their works are based on appliance-level agent negotiations but do not use rule-based policies, and agents can define just one collective goal. Sampaio et al. [30] use negotiation among agents to maximize the number of restored loads, so the agents of the adaptive protection system (APS) reset the protection settings after changes occur in the network topology and state of distributed generations (DGs). The APS can change the current setting group and, if necessary, calculate new settings and sends them to the relays to ensure the accurate coordination of protection systems. The authors divided the system into three levels: device layer, control layer, and decision layer, and reliability in medium-voltage achieved, but they are not considering houses and their appliances in the adaptation plan. Their Self-healing plan works when all agents are well, but they do not present any repairing plan if an agent, for example, a DCA agent, goes off. Huang et al. [36] proposed a multilevel dynamic master–slave control strategy via two-level dynamic leaders to realize the resilience-enhanced power management of networked microgrids. The dynamic feature enhances the system resilience under the pinned terminal distributed energy resources outages and communication failure. Still, they need a plan for the loss of each level leader. El Zerk et al. [37] introduce the concept of collaborative microgrids (CMGs) with shareable resources, allowing households to cooperate and communicate intelligently to establish a reliable and proprietary CMG. It focuses on the microgrid's performance, such as dynamically determining power exchange, reducing losses, allowing households to contribute energy, tolerating the availability of one or more MGs, decentralizing storage, and having programming flexibility. But in their model, when MG agent fails, no coordination is possible between the agents. Hamidi et al. [38] propose a novel multi-agent model-based intelligent control scheme to balance AC/DC load demands and renewable energy sources. It consists of a three-level hierarchical system. If one agent in their hierarchical model fails, they can not reorganize it again, so coordination in that section is impossible. Jiménez et al. [39] proposed a framework to reduce peak loads, detect overloading effects, and provide demand forecasting based on autonomous and distributed systems, but without the self-organization of agents. Blechmann et al. [40] present an open-source platform based on the FIWARE software framework and other open-source components to perform experimental cloud control on two use cases from the smart building and smart grid domains. The effectiveness of such controls depends on delays in the communication network, cloud processing, and possible other delays. They did not consider the self-organization of agents and collaboration for energy exchange between agents. So far, no model has been presented for adapting the control system of multiauthority microgrids when a component fails and automatic return of the repaired component to the system. As a result, improving the reliability and availability of this system is the primary goal of this article. This research presents a Decentralized Control Architecture for a set of Multi-authoring microgrids (DCAM) that is self-adapting and self-repairing. The main contributions include the following: (1) Proposing an architecture for microgrid self-organization to increase resilience in unfavorable conditions and make it possible to increase energy exchange. (2) Providing macro and micro policies for

controlling a large-scale distribution network with multi-authoring at different levels. (3) Defining different control messages in the DCAM and how to control agents for adaptation purposes. (4) Presenting scenarios for self-adaption of control agents in different levels and approve them by simulation. The proposed architecture can be used in other similar areas of ambient intelligence, such as cyber-physical systems, IoT systems, and smart city systems for control and self-adaptation of these systems [27, 31]. For ambient intelligent environments such as smart houses, the DCAM can adapt house environments depending on different conditions and user policies (depending on their time, interests, and activities). Also, DCAM by distribution architecture is scalable and efficient by increasing its holons(like smart houses).

3 Proposed architecture

The main requirements of any decentralized control system are integration, the ability to change local and global control units' goals without stopping it, self-adaption with limited information, and high reliability. The proposed DCAM, a holonic architecture, is shown in Fig. 2a. In the proposed DCAM, holons correspond to micro-grids. They can work together to achieve mutual goals, thus being able to adapt autonomously to changes in their environment. Holonic architecture increases reliability because each holon can complement the other holon and meet its needs, and depending on environmental conditions, holons can be detached from their super holon (a holon of which some holons are members) and attached to another holon that characteristics of its members are complementary to the characteristics of the holon (maximum resource utilization).

The DCAM architecture is designed using a multi-agent model because it is commonly used for self-organizing systems with decentralized and dynamic control [33]. As a result, each holon's components are considered an agent, and each agent performs a specific DCAM task autonomously and cooperates with the other agents in everyday tasks. In DCAM, due to its scale and different layers, heterogeneous agents are used, which have different computational power according to the processing volume. In the following, we introduce all parts of DCAM. According to the left of Fig. 2b, at the lowest level of DCAM (level 1), for each device that consumes or generates electricity, called a managed device (MD), we considered a device control agent (DCA). As shown in Fig. 2c, each device has a DCA that sends its load states to DCA periodically. The DCA evaluates the condition in input instruction from its HOCA, and then if the adaption is needed, it generates the new instruction to adapt the devices. Devices have four types, consumers like lamps, batteries, smart meters, and distributed generators like solar generators. For each consumer device i , we compute the consumption load ($Con_i(t)$) as follows:

$$Con_i(t) = CCon_i(t) \times DataCon_i(t) \quad (1)$$

Where $DataCon_i(t)$ is a load consumption of device i at time t that DCA_i gathered from device i via Load State messages. DCA_i with a control parameter ($CCon_i(t)$) that send it by instruct messages, controls device i consumption.

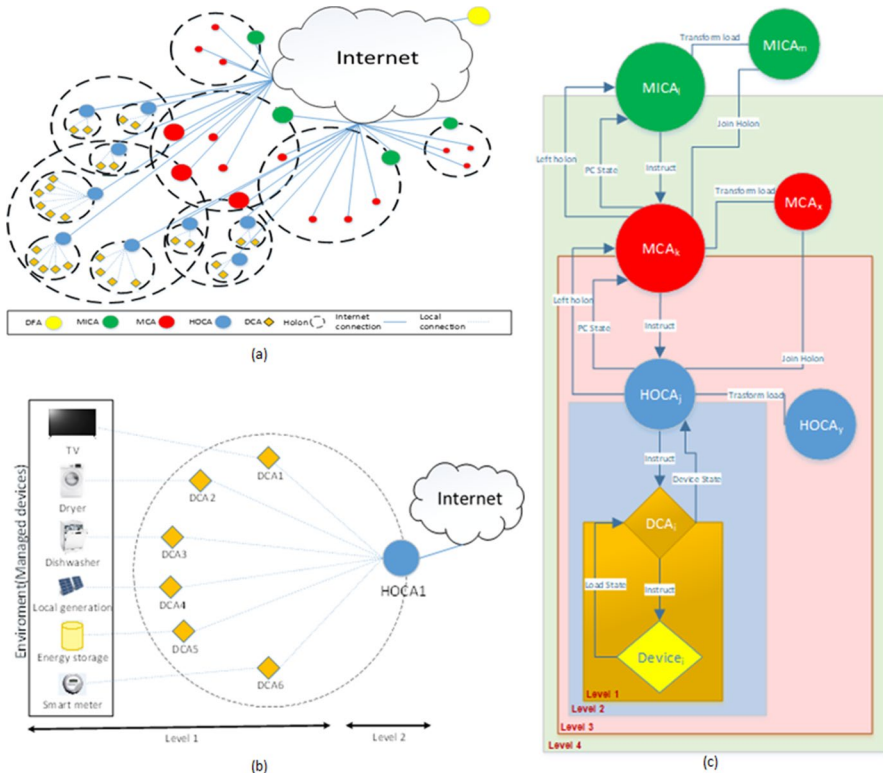


Fig. 2 a DCAM holonic architecture. b Example of a house holon in DCAM. c 4-level distributed holonic control paradigm of the DCAM architecture

$$CCon_i(t + 1) = \begin{cases} \min(1, CCon_i(t) + step(i)) & Con_i(t) \geq maxLoad \\ & \text{and } priority(i) \geq p \\ & \text{and } CCon_i(t) = 1 \\ \max(0, CCon_i(t) - step(i)) & Con_i < maxLoad \\ & \text{and } CCon_i(t) = 0 \\ CCon_i(t) & Else \end{cases} \quad (2)$$

$$step(i) = \begin{cases} 1 & i \text{ is a regular device} \\ 0.001 & i \text{ is a dimmer device} \end{cases} \quad (3)$$

For all load consumption of a house at time t , we have:

$$Con(t) = \sum_{i \in Consumers} Con_i(t) \quad (4)$$

For each locally distributed generator i , like photovoltaic (PV) systems or wind turbines, we have:

$$Gen_i(t) = CGen(t) \times DataGen_i(t) \quad (5)$$

Where $DataGen_i(t)$ is a load generation of generator i at time t that DCA_i gathers from generator i via Load State messages. DCA_i uses a control parameter ($CGen(t)$) for controlling the generator i output that it sends via instruct messages. For all load generation of a house at time t , we have:

$$Gen(t) = \sum_{i \in Generators} Gen_i(t) \quad (6)$$

$$CGen(t+1) = \begin{cases} 1 & Con(t) < Gen(t) \\ 0 & Else \end{cases} \quad (7)$$

For all local batteries, we compute the input load as:

$$BtIn(t) = CBt(t) \times (Gen(t) - Con(t)) \quad (8)$$

DCA_i , via instruct messages, sends a control parameter ($CBt(t)$) for controlling Battery input.

$$CBt(t+1) = \begin{cases} 1 & Con(t) \geq Gen(t) \\ & \text{and } Con(t) < Bt(t) \\ 0 & Else \end{cases} \quad (9)$$

We define the Current battery's charge as follows:

$$Bt(t) = \begin{cases} MC & BtIn(t) + Bt(t-1) > MC \\ 0 & Bt(t) + BtIn(t) < 0 \\ Bt(t-1) + BtIn(t) & Else \end{cases} \quad (10)$$

Smart meter controls loads from the main grid by a control parameter ($ControlSm(t)$) that sends it by its controller.

$$Sm(t) = CSm(t) \times Con(t) \quad (11)$$

So $Sm(t)$ shows consumption loads from the main grid. We define control parameters as follows:

$$CSm(t+1) = \begin{cases} 1 & Con(t) > 0 \\ & \text{and } Con(t) \geq Gen(t) \\ & \text{and } Con(t) \geq Bt(t) \\ 0 & Else \end{cases} \quad (12)$$

At the house level (Level 2), each house/apartment is a holon and is represented by the house control agent (HOCA) and manages the DCAs of that house/apartment (that is added directly by the building manager). The interactions between the HOCA agent and the members of the house holon are via the wired or wireless local area network of the building (Fig. 2b). In each predefined periodic time, DCAs throw a Device State message, send its Load information (Load State) to HOCA. However, it can decide on controls and adaptation of MDs via control instructions that send via instruct messages to MDs. At the microgrid level (Level 3), HOCAs of houses in a region of the city can join a maximum of k microgrid (holon) from different companies in that region (to increase reliability). A microgrid control agent (MCA) controls the microgrid. In DCAM architecture, the formation of holons at this level is self-organized and based on the limitations of the geographical region and the facilities of holons. Interactions between Level 2 and higher agents can be based on Internet protocols and in the context of a microgrid or Internet communication network. The HOCA sends aggregate information of its MDs (PC State) at time t ($PCState(t)$) to each microgrid it has joined. $PCState(t)$ contains $Con(t)$, $Gen(t)$, $Bit(t)$, $Sm(t)$, t , and $HouseID$. A house to join or disjoin a microgrid must do the process of joining Holon and leaving Holon, which we explain in section IV. Suppose a house in the microgrid needs more loads. In that case, it requests from microgrid MCA to get more loads from its neighbors, and MCA introduces selected houses and houses negotiated with them through the transform load process, which we explain more in section VI. At the city level (level 4), there are Microgrid interchange control agents (MICAs). These agents are responsible for controlling the exchange of power between microgrids. Each MICA can only serve a specific geographic region, but in one geographic area, there can be several MICAs owned by different private companies. In the DCAM architecture, microgrids can join MICAs autonomously based on the geographic region. Similarly, MICAs can control power exchange between microgrids in cities in one part of the country. The negotiations between members are similar to negotiations of microgrid members. This hierarchy can continue to control the power of parts of a country. In this paper, we described DCAM architecture at the city level and evaluated it. In DCAM architecture, unlike DCAs, which are all statically organized at the level of a house, higher-level control agents can be dynamically organized in holons, which we call dynamic agents.

4 The process of formation and membership in holon

As mentioned, each level 2 control agent (HOCA) represents a holon that has statically defined members (DCAs) that are defined by the house manager. Any control agent at level 3 or higher can form a holon with dynamic agents to any number so that its members increase or decrease at runtime. The number of members of these holons can even be zero or at least have a static number, for example, defined by the microgrid manager for MCA. At the start of each control agent at level 2 or above, that agent sends its ID, geographical region code based on urban divisions, and level to the city directory facilitator agent (DFA). DFA is the agent that is responsible for maintaining the ID, level, and area of dynamic city agents in DCAM, and the ID of

each control agent is saved uniquely after creation in DFA. To begin the registration process, each level 3 or higher dynamic control agent receives a list of lower-level control agents in their area from the DFA of that city. For control agents who are not members of the holon, an invitation message to holon is sent includes the amount of production, consumption, and storage. Lower level control agents will receive and review the invitation of the higher-level control agent and if they are free or only members of the k-1 holon, and if membership in that holon following their policy, they would send back the agreed message and also inform of their status (including production, consumption, and storage). Each higher-level control agent, after receiving a response from the lower level control agents, if their response is following its policies, send back its final agreement message to them so that they become a member of the higher-level holon and henceforth the new member will be able to share or use holon resources. Figure 3 shows the membership process of a HOCA control agent named H1 in a microgrid with a control agent called M1. The membership process is repeated periodically so that control agents can select the best holon that is consistent with their policy based on the changes in holons.

Each holon member is required to periodically submit their status information including production, consumption, and storage to the holon dominator. If the information is not sent to the dominator by the period, the dominator will expel the member from holon, and if the expelled member re-wishes to join holon, it will have to start the membership process from the beginning. Also, if a control agent notices that holon does not comply with its policies with the monitored information from holon, it can separate from the holon by not sending its information periodically to holon's dominator. In addition to periodically monitoring member information, the holon dominator receives the emergent messages of holon members, reviews and designs adaptation operations for holon, and suggests them to holon members for execution; e.g., orders to exchange or reduction of energy consumption depending on the planned amount.

5 Architecture and tasks of control agents

The control agents introduced in the second section are all self-adaptive, meaning they can change their behavior autonomously in response to changes in their operating environment. At the lowest level of DCAM (i.e. level 1), the operating environment of each DCA control agent is a managed device and the physical environment around that device. The operating environment of the HOCA control agent (corresponding to level 2 of the DCAM architecture), DCAs under the management and control of that control agent, and the microgrids to which they belong. The operating environment of the dynamic control agent, MCA (corresponding to level 3 of the DCAM architecture), its holon members (i.e. microgrid houses), the dominators of the holon that joined, and the adjacent MCAs in the holon. At the highest level, the operating environment of the dynamic control agent, MICA (corresponding to level 4 of the DCAM architecture), its holon members, the microgrids. Control agents in DCAM architecture perform their tasks based on the MAPE-K model [34] (Fig. 4), using the five components: monitoring, analysis, design, execution, and a shared

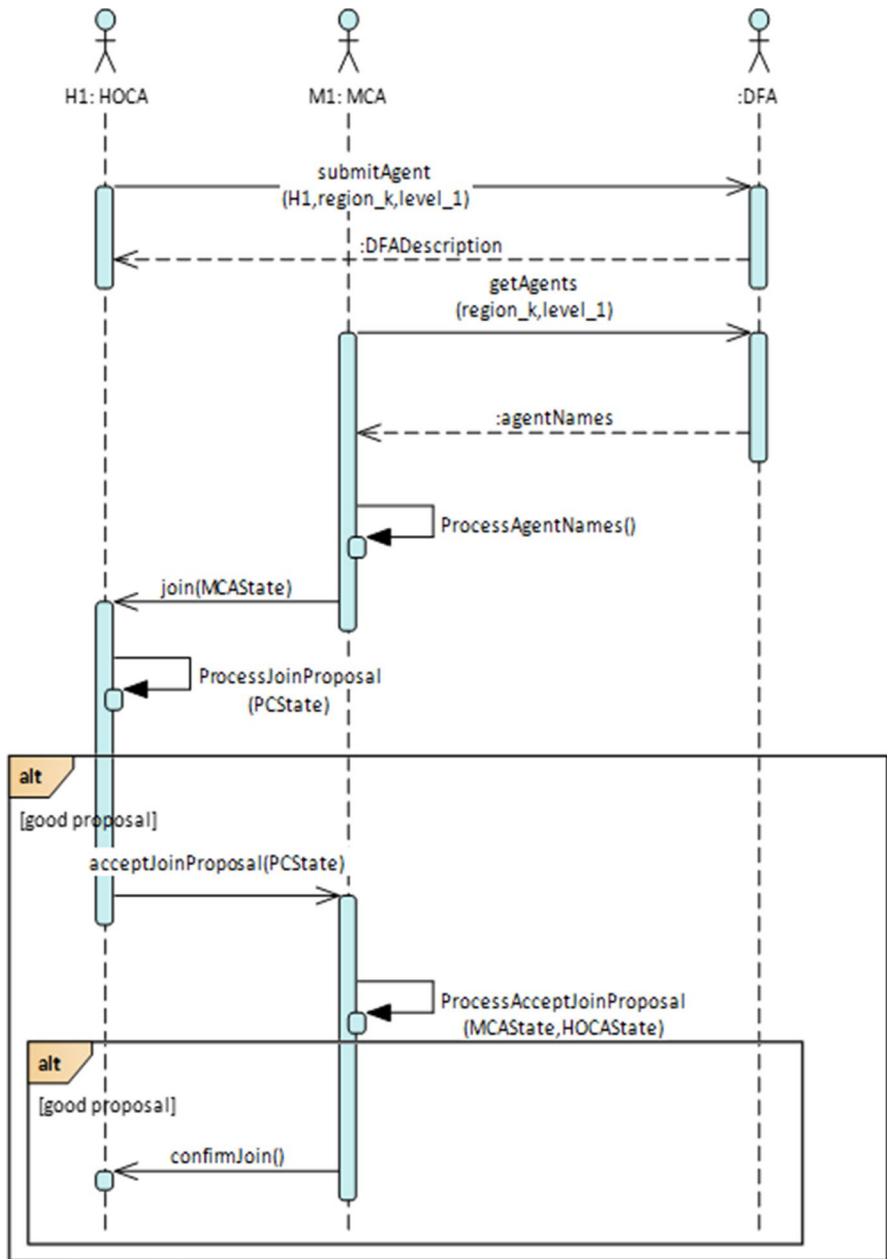


Fig. 3 The process of joining a HOCA control agent named H1 in a microgrid with a control agent called M1

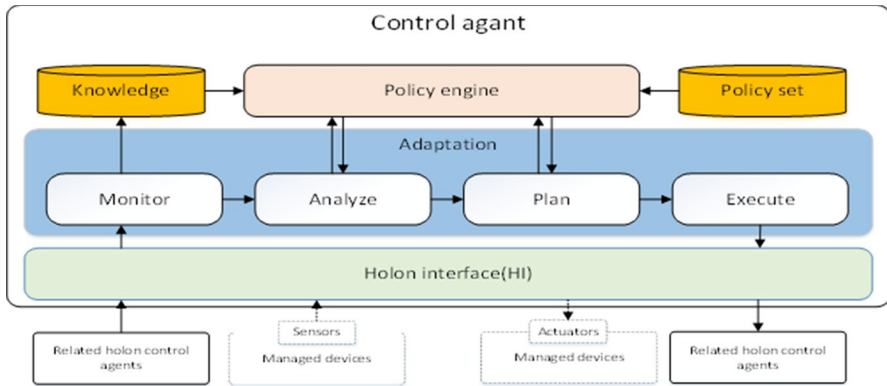


Fig. 4 Control agent architecture

knowledge memory to interact with each other and design adaptive behavior. Analysis and design activities are performed using the policy engine component based on the policy set defined by the control manager. The holon interface (HI) for receiving and sending control agent messages to related other control agents and in DCA control agents, also, it is for connection to the managed device. The DCA control agent receives the managed device monitoring information from its sensors and, if required to adapt the managed device, sends the adaptation command messages to the managed device actuators.

5.1 Adaptation cycle

The adaptation cycle algorithm in DCAM architecture is presented in Fig. 5. The action method in each control agent is called upon receiving a new message through the holon interface. The UpdateStatus method monitors the component and saves the types of messages that the component receives in shared knowledge. The AnalysisStatus method is responsible for the analysis component, which, depending on the type of the last message received, performs a proportional analysis using the policy engine to determine the need for adaptation, and the result of the requirements for adaptation is saved in shared knowledge memory. If there is a need for adaptation, the Plan method (design component) for designing adaptation is called based on shared knowledge information, and its output is a list of commands that are given to the Execute method to execute.

5.2 Monitor component

In the monitoring component of each control agent, messages received from other control agents or MDs are checked and its results are saved in the shared knowledge memory and the action method is called. Another task of the monitoring component is to collect the status information of holon members and periodically send

```

void action() {
    lastMessage = myAgent.receive();
    UpdateStatus(lastMessage);
    boolean NeedAdaption=AnalysisStatus();
    if(NeedAdaption){
        List<Instruction> instructs =Plan();
        Execute(instructs);
    }
}

void AnalysisStatus() {
    NeedAdaption=false;
    policyEngine.InsertFact();
    switch ( lastMessage.getType())
    {
        case t1:
            if(policyEngine.GoodCondition())
                return;
            break;
        case t2:
        case t3:
        case t5:
            if(!policyEngine.canDoInstruct())
                return;
            break;
        case t4:
            if(!policyEngine.isGoodRequest())
                return;
            break;
    }
    NeedAdaption=true;
}

```

Fig. 5 Adaptation cycle algorithm in DCAM architecture

summarized them to the holon dominator of which it is a member. MCA and MICA control agents receive a list of their region control agents from the DFA agent with their monitoring components.

5.3 Analysis component

The analysis component of each control agent examines the fulfillment of the policies defined by the manager of that control agent using the policy engine and based on the shared knowledge memory information and the policy set defined in that control agent, and the result that needs to adaption saves to knowledge memory. In the action method, if the adaptation is needed, the adaptation design component is called (Fig. 5). In the AnalysisStatus() method, in the beginning, the policy engine detects new conditions by InsertFact() method. In the switch case, the agent evaluates the last message type defined in the I subsection. If it is a monitoring message(t1), the

policy engine checks the condition by `GoodCondition()` method, and if not ok, then the agent must design an adaption. Otherwise, if the message type is an instruction or emergency request or load transfer message, the agent policy engine by the `canDoInstruct()` method checks if it could do the receive instruction/request, and if the design adaption and else skip it. If message one of the membership request types to the holon, the policy engine checks that by the `isGoodRequest()` method, and if the condition is ok, the agent needs adaptation.

5.4 Design component

If the result of the execution of the analysis component shows that the holon or the managed device is not in good status, the design component is called to design an adaption plan according to the items specified by the analysis component to adapt the control agent. The adaptation plan consists of a set of instructions for other control agents. The adaptation plan that is the division of tasks between members of the holon, depending on their situation, is solved as an optimization and multi-objective problem, like [11, 13, 15, 22]. If the holon member needs with the help of other control agents of the holon, an emergency request message is sent to the dominator of the holon. For example, if a holon dominator receives an emergency request load L_E from its j -th member of its holon, $HOCA_j$ control agent, where its policy is to accept all requests, The holon dominator could use the following utility function:

$$\text{maximize } \sum_{i=1}^n w_i \times x_i \quad (13)$$

subject to $\sum_{i=1}^n w_i \times x_i \leq L_E$ and $0 \leq x_i \leq 1$ and $i \neq j$ and $w_i = SC_i - L_i - M_i$. SC_i , L_i , M_i , w_i , x_i are the i -th member's charge on its storage, load, minimum storage load, available load, and dedicated power for exchange respectively.

5.5 Executor component

To execute the adaptation program generated by the design component, the executor component is called by the action method, and the result is the delivery of messages to the holon interface to send.

5.6 Policy set

The policies of each control agent defined by its holon manager before or during execution without the need to stop the execution of the control agent. Policies are in the form of a set of event-based rules (ECA). The policies in each holon are the local policies of that holon, but the set of policies in DCA control agents are for managing the device by those control agents as defined by the house owner.

5.7 Policy engine

The task of the policy engine is to detect events in the system that need to be adapted. The act of discovery is performed by examining the preconditions of each policy, and as a result of its establishment, the instructions of that policy are executed, and the result of execution, which specifies the requirements for adaptation, is saved in shared knowledge.

5.8 Holon interface

Sending and receiving messages is done through the holon interface of the control agent. Managing the receiving and sending queue of control agent messages and validation of the received message (such as identifying and validating the sender) and registering the control agent region in the DFA agent are other tasks of this interface.

5.9 Types of messages

The types of adaptive messages are as follows: (1) Monitoring (t1): Contains status information for a holon or managed device. (2) Command (t2): An instruction from a holon dominator in which the control agent is a member to change the status of the control agent or its environment. (3) Emergency request (t3): A request from the control agent to the holon dominator to receive resources. (4) Membership (t4): is a request or response message in the holon membership process. (5) Load transfer (t5): is a request or response message in the load transfer process. The DCA controller sends t1 messages to inform the status of the MD under its control. HOCA and MCA control agents send t1 messages to a higher-level holon dominator to inform the state of holon they are dominated. The MICA and MCA also send t1 messages to all their holon members to inform the general status of the holon. The t1 message is sent periodically by each control agent and when a control agent receives a new monitoring message, first validates and saves it in its shared knowledge memory through the HI component and then calls the action method (Fig. 5). Figure 6 shows the types of messages that can be transmitted between control agents at different levels of the DCAM architecture.

5.10 The holonic role

The main difference between control agents in DCAM architecture is their holonic role. The static DCA control agent is only responsible for managing a specified MD. HOCA Control Agent Manages DCA control agents in a house. Each HOCA control agent can automatically decide whether or not to join the microgrid that offers membership. However, DCA control agents are predefined members of a level 2 holon, and can't join other holons. MCA control agents are similar to HOCA control agents, except that they can recruitment and their members are joined dynamically. MICA control agent because it is at the highest level,

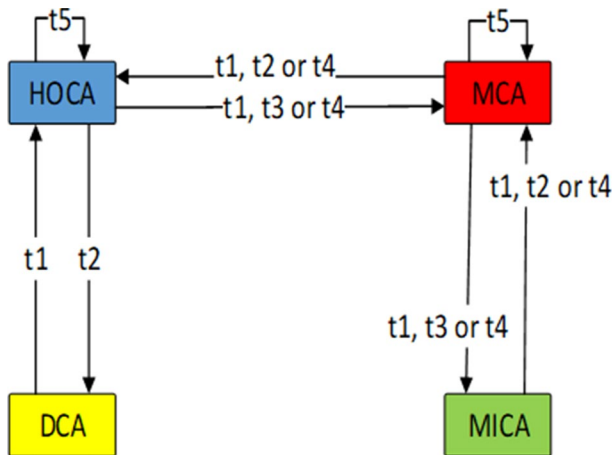


Fig. 6 Types of messages that are exchanged between control agents

cannot become a member of another holon and in other respects is like MCA. It should be noted that in each of the control agents DCA, HOCA, MCA, and MICA, a set of policies are embedded that corresponds to the managerial responsibility of that control agent. The local manager of each control agent can view and manage that agent's policy set through the user interface. The local manager is the house owner at levels 1 and 2, the microgrid operator at level 3, and the regional power distribution manager at level 4. If a holon adaption instruction that based on holon policy conflicts with agent local policy, priority will be given to holon's dominator instruction.

6 Designing adapt program

According to the DCAM holonic architecture, the design of adaptation of each holon is performed by holon dominator's design component using its policy engine, and the adaption plan is suggested or instructed to its members. Each member independently receives instructions and suggestions from its holon dominator and can design an adaption program and inform it to its holon members or its managed device. With the adaptability of control agents, it is possible to react quickly to the events of the main grid and the unpredictable environment of a control agent. In case of power shortage or failure in a part of the smart grid, it will be possible to design an adaptive program to balance the load distribution and increase the reliability and availability of the network and reduce blackouts and power fluctuations (self-repair). Also, by changing the goals of the system at any level without the need to stop the smart grid, it will be possible to adapt the behavior of the smart grid to new goals. In case of failure of any part of the smart grid, the smart grid is reorganized, repaired, and after fixing the failure member at runtime, the network with a new adaptation, again adds the repaired member

to itself. The following are the adaptations that can be made at different levels of DCAM.

6.1 Adaptation cases in level 1 of DCAM architecture

At level 1 of the DCAM architecture, each DCA control agent can control the amount of power consumed by the device under its control or the transmission of power from input sources in a house based on policies in the DCA controller or commands received from its holon dominator (i.e. HOCA controller). For example, due to the need to save power, the house owner defines the blackout policy during the day for the lighting control agent (i.e. DCA controller). In this case, the DCA control agent commands the lamp to turn off the light when it is on during the day. In another example, the HOCA controller of the house at peak consumption instructs the members of the house (i.e. level 1 of the DCAM architecture) to reduce the consumption, so the instruction to reduce the brightness during these hours is sent to the lamps by the DCA controller.

6.2 Adaptation cases in level 2 of DCAM architecture

The HOCA control agent, which dominates a house, through the day and night, selects the power supply of the house according to the policy set which is defined in that control agent. For example, if the output of a solar energy source in the house is less than a γ threshold, it switches to other sources such as batteries or the main grid. In the absence of any of them, the power applicant's control agent sends an emergency request for power to the dominators of the holons (i.e., the dominators of the microgrids whose house is a member). Also, according to its policy set, the house control agent can instruct the household appliances to reduce consumption so that all non-vital appliances in the house can reduce their consumption. After repairing the power shortage problem, the HOCA house control agent instructs the house appliances to use electricity normally. For another example, the HOCA control agent can select a microgrid following its local policy from among the local microgrid membership offers, or leave the microgrid if continuing to subscribe to a microgrid does not comply with its local policy. If the HOCA control agent has a proposal from the holon neighbor to supply the load, if the supply is in line with his local policy, he can respond positively to the neighbor's request and transfer it from his power storage sources to the neighbor's house (Fig. 7).

In the event of a DCA control agent failure for an input power load to the house, the HOCA control agent removes the DCA control agent statistics from its sources and first designs and performs the appropriate adaptation accordingly and then notifies the dominators of the registered holons of the change in the load input of the house. For example, if the storage source fails, it can instruct DCA to reduce the power consumption of other appliances until that power supply is repaired.

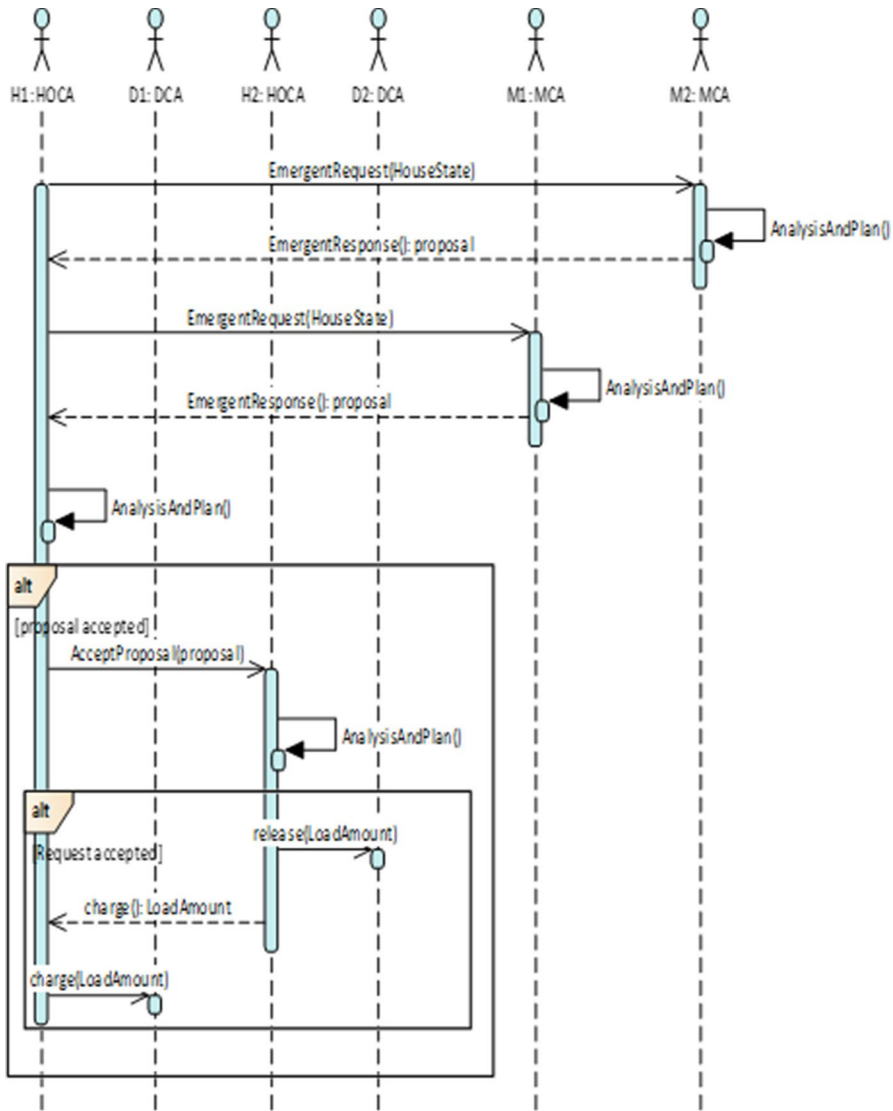


Fig. 7 The process of adapting home resources due to shortage of H1

6.3 Adaptation cases in level 3 of DCAM architecture

At level 3, the MCA controller responds to the demand of the region houses (i.e. HOCA control agents, dominating the house holons) for membership in the microgrid, and if a house meets the defined requirements, it accepts the house membership. Also, if a microgrid member house violates or breaks the membership requirements, the MCA control agent will remove it from its membership list and it becomes a free member. If from the MCA control agent, which controls a microgrid,

a house holon member requests a power supply due to short of resources (Fig. 7), according to the amount of storage of the microgrid members, it has designed an adaptation and suggests the amount of electricity supply from each holon member to the house that electricity applicant. The applicant's HOCA control agent can select the best proposal from the various microgrids of which it is a member and meet its needs. As a result, in addition to supplying power to the applicant's house, the optimal power supply is supported by offers received from various microgrids in the DCAM architecture. If the consumption in a microgrid exceeds the allowable limit or receives a command from a higher level holon dominator, the microgrid control agent instructs holon members (i.e. houses) to reduce power consumption. If the MCA controller observes a shortage of power supplies in its microgrid or has an emergency request from a microgrid member and is unable to meet that house's need from other members, request a power supply from the holon dominators it belongs which will have a process similar to that of the HOCA control agent resource shortage case.

6.4 Adaptation cases in level 4 of DCAM architecture

At level 4, the MICA control agent monitors the status of its metropolitan region microgrids and try to meet their needs from other MICA control agents in other metropolitan areas. The operation of the MICA control agent is similar to the MCA control agent, except that the policies defined at that level are higher and the holon members are MCA control agents. The MICA controller can, in case of cooperation policy, order the members of the main grid to cut off the consumption of power from the main grid to act as an island so that the other part of the city can use the main grid or can export power stored in its microgrids to another part of the city.

6.5 Holon dominator failure

If the dominator of a holon is failed, for example, it is shut down or has a technical defect, it is removed from members of the holons of which it is a member, and its members are also removed from that holon as a free member. If repaired, the following processes will be performed at each level: A) At Level 2, if a HOCA control agent is repaired, it updates its predefined DCA control agent statistics and applies for re-membership in region microgrids. B) At levels 3 and 4, if a dominator is repaired, it introduces itself as a proposed dominator to the lower level control agents of the region and starts the recruitment process and applies for membership in higher-level holons.

7 Implementing dcam architecture

Using the Simulink simulator, we have simulated all the electrical equipment in the house and the devices for generating, storing, and consuming electricity in the house, and all the control agents, membership process in holon and relationships

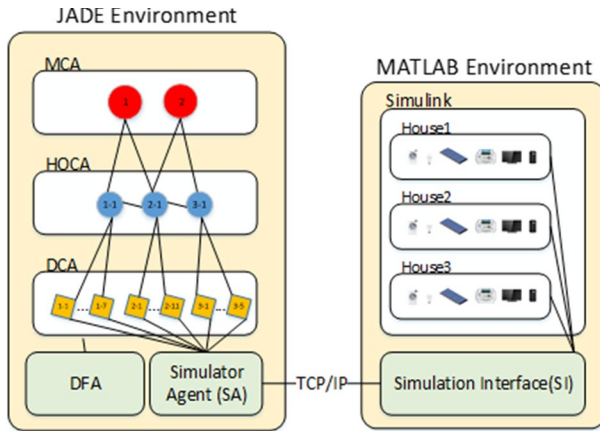


Fig. 8 DCAM simulation and implementation model

Fig. 9 Policy implemented in the form of a rule using the Drools tool

```

rule "Adapt restore holon consume"

when
    $f:ControllerFact(isGoodCondition==true)
    and
    $s:PCState(load<$f.maxLoad+0.01)
then
    $f.needsChangeUp=true;
    $f.isGoodCondition=false;
end

```

and hierarchies of holons implemented in the JADE environment and creating the relationship between JADE and Simulink using socket programming technology based on a publish-subscribe architecture (Fig. 8).

The connection between the JADE environment and Simulink is created using an agent called simulator agent (SA) in the JADE environment and using an interface called Simulink interface (SI) in the Matlab environment. At each step, the SA agent receives the monitoring or adaptation commands request from the DCA control agents and sends them to the SI interface via the network connection. Depending on the type of message received, the SI interface will configure the device in Simulink if the message is adaption command, otherwise, it will receive the monitoring information of the requested devices from Simulink and send them to the SA agent in JADE. The Drools tool is used to define and implement policies in control agents. For example, the code corresponding to the “holon power recovery” policy can be seen in Fig. 9.

This policy is to recognize the need to return consumption to its previous state after reducing consumption adaption. In this policy, the title of the policy comes after the rule keyword, the conditions (executive preconditions) of the policy comes after the keyword when and the executive operation of the policy comes

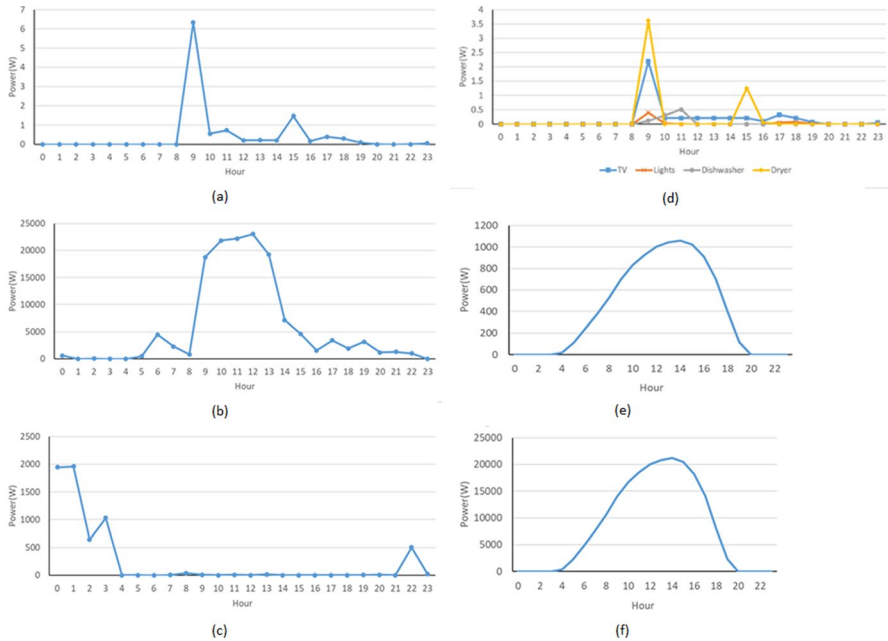


Fig. 10 **a** Power consumption chart of type 1 house in 24 h. **b** Power consumption chart of type 2 house in 24 h. **c** Power consumption chart of type 3 house in 24 h. **d** Power consumption chart of type 1 house devices. **e** Solar energy production chart of type 1 house. **f** Solar energy production chart of type 2 house and type 3 house

after the keyword then. The event that leads to the evaluating of the policy situation is the receiving of a new message by the control agent. According to this policy, if the holon conditions are normal and the amount of load consumption is less than the maximum load defined for the holon, the adaption operation in the then section will be executed to return consumption to the previous state. In the Simulink simulator environment, three types of buildings are defined that each one is both producer and consumer, with different consumption patterns and the ability to store power in batteries.

8 Evaluation of dcam architecture

A computer with a Corei7 4790k processor, 16GB of RAM, and 8 logical processing cores was used to evaluate DCAM. The simulation was performed for 24 h using the Australian Intelligent Grid Database [35]. Also, the data set in the MATLAB smart grid example is used to simulate the load of solar energy produced. In the evaluation, a microgrid with five houses type 1 (including control agents for managing lighting appliances, TV, dishwasher, dryer, solar panel, storage, and smart meter) and three, type 2 houses (including control agents for managing the device general consumption, solar panel, storage, and smart meter) and a type 3 house (including

Table 1 The three house types specifications

Type	Load pattern	Max load (W)	Storage battery (W)	Generation type and capacity (kW)
1	Fig. 10a	6.336	90	Solar, 1.1
2	Fig. 10b	23040	4000	Solar, 21
3	Fig. 10c	1963.002	4000	Solar, 21

control agents for managing general consumption devices, solar panel, storage, and smart meter) for a total of 51 control agents. The amount of power consumption, without adaption, for type 1 house is shown in Fig. 10a, for type 2 house is shown in Fig. 10b, and for type 3 house is shown in Fig. 10c. The amount of power consumption of type 1 house devices is shown in Fig. 10d.

Figure 10e shows the amount of solar energy produced in a type 1 houses and Fig. 10f shows the amount of solar energy produced in type 2 houses and type 3 houses.

The battery capacity and the specification of three type houses was shown in Table 1, and the initial storage capacity of the houses when starting the simulation was 90 watt.

24-hour simulations of DCAM architecture at levels 1, 2, and 3 are evaluated. For this purpose, three microgrids numbered 1 to 3 were defined with control agents MCA-1, MCA-2, and MCA-3 so that each has a fixed member. Fixed members cannot change their microgrid and cannot become a member of another microgrid. While free members will have the possibility of dynamic membership in a maximum of two microgrids. The HOCA-11 control agent is of type 1 house, fixed member of microgrid 1, the HOCA-21 control agent is house type 2, fixed member of microgrid 3 and the HOCA-31 control agent is house type 3, fixed member of microgrid 2. HOCA-12, HOCA-13, HOCA-14, and HOCA-15 are free, type 1 house control agents and HOCA-22 and HOCA-23 are free, type 2 house control agents.

9 Discussion

At level 1 of the DCAM architecture, lighting adaption with the HOCA-11 control agent is evaluated. The amount of light consumption is shown in Fig. 11a. To prevent energy loss, the policy defined for the lighting control agent is to turn off the lights between 6 AM and 6 PM. As can be seen in Fig. 11b, the light consumption after adaption is zero between 9 and 17 o'clock, but the light consumption is on at 18 o'clock.

In level 2 of the DCAM architecture, maximum power consumption in an hour is defined as 5 watts per hour for the HOCA-11 control agent. At 9 a.m., the amount of consumption exceeds 6Wh (Fig. 10a). As a result, the HOCA-11 control agent instructs to reduce consumption by sending a message to devices that can reduce consumption (dryer and lighting). As a result, the dryer program changes and continues to work with a program with less consumption than allowed. At 10 a.m.,

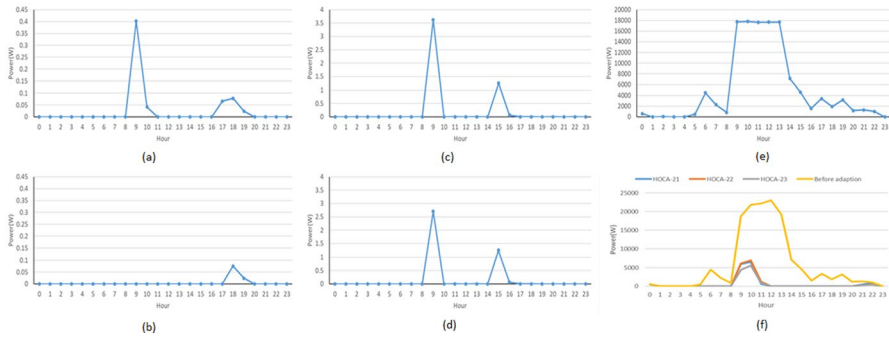


Fig. 11 **a** House lighting power consumption under control of HOCA-11 agent. **b** House lighting power consumption under control of HOCA-11 agent after adaptation according to the lighting off policy between 6 AM to 6 PM. **c** Dryer power consumption under control of HOCA-11 agent. **d** Dryer power consumption under control of HOCA-11 agent after adaptation according to the maximum house power consumption policy. **e** Dryer power consumption under control of HOCA-11 agent after adaptation according to the maximum house power consumption policy. **f** Type 2 houses power consumption from the main grid

consumption is lower than the allowable limit, as a result, the consumption pattern changes to the machines before adapting with HOCA-11 control agent command to devices, so dryer and lighting continue to work normally. Figure 11c shows the amount of dryer consumption before adaptation and Fig. 11d shows the amount of dryer consumption after adaptation. Also, the HOCA-21 control agent house consumption after adaptation with the policy of not consuming more than 18,000 Wh between 9 and 13 o'clock is shown in Fig. 11e. According to the view of HOCA-21 house consumers in general, the reduction of consumption between 9 and 13 o'clock has been done by applying the reduction of the total house consumption coefficient by the HOCA-21 control agent and consumption is backed to normal by the HOCA-21 control agent after 13 o'clock. This also indicates the control of consumption in critical situations, so that if the main power grid is cut off from the microgrid, it is possible to reduce consumption based on internal sources. At the microgrid level (Architecture Level 3), two general policies are defined for house control agents to join microgrids: (1) Unconditional acceptance of membership offer (type 1 houses). (2) Accept membership offer if the total microgrid storage volume is more than 1000 watt and leave holon if the total microgrid storage volume is less than 500 watt (type 2 houses). Table 2 shows the membership process of house control agents in holons by execution time. As can be seen, the control agents of type 2 houses become members of or separate from the microgrids, depending on the amount of storage available in the microgrids. This helps them proper utilization of most of the energy resources available in each holon.

Type 1 house control agents, on the other hand, do nothing to change their microgrids after determining their two microgrids. Another adaptation that occurs at this level because of a lack of energy resources in the house and the house request from the dominator of member microgrids to supply it. Type 2 agents, due to the need for high consumption in the morning and the lack of sufficient storage and production

Table 2 Membership process of house control agents in holons

Time(s)	Agent	Parent1	Parent2
0	HOCA-11	MCA-1	
0	HOCA-21	MCA-3	
0	HOCA-31	MCA-2	
120	HOCA-14	MCA-1	
120	HOCA-13	MCA-1	
120	HOCA-12	MCA-1	
120	HOCA-15	MCA-1	
120	HOCA-12	MCA-1	MCA-2
120	HOCA-13	MCA-1	MCA-2
120	HOCA-15	MCA-1	MCA-2
120	HOCA-14	MCA-1	MCA-2
19140	HOCA-22	MCA-2	
19140	HOCA-23	MCA-2	
19920	HOCA-22	MCA-2	MCA-3
19920	HOCA-23	MCA-2	MCA-3
33180	HOCA-22	MCA-2	
33180	HOCA-23	MCA-2	
36960	HOCA-22	MCA-2	MCA-3
36960	HOCA-23	MCA-2	MCA-3
38100	HOCA-23	MCA-2	
38100	HOCA-22	MCA-2	
43140	HOCA-22	MCA-2	MCA-3
43140	HOCA-23	MCA-2	MCA-3
76920	HOCA-22	MCA-2	
76920	HOCA-23	MCA-2	

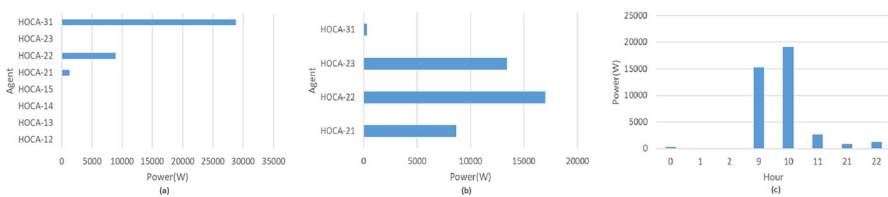


Fig. 12 **a** The amount of load provided by the house control agents. **b** The amount of load received by the house control agents. **c** The amount of load exchanged between the houses during the simulation time

of energy, have made an emergency request to the dominator of the microgrids that are members to provide the resources from the microgrid if possible. The amount of load received by these control agents is shown in Fig. 12b and the control agents that supplied the load are shown in Fig. 12a and the amount of load exchanged during the simulation is shown in Fig. 12c. As can be seen, type 2 agents are the major receivers of load exchanged. The amount of consumption from the main grid in type

2 houses is shown in Fig. 11f, and we can see that the consumption of Type 2 houses is much lower than when there is no adaptation operation.

Compared to [36–39], our results show that DCAM is resilient to leader agent failure and can reorganize itself to a good state for collaboration and energy exchange. Moreover, our agents can join multiple holons, have a better chance of energy exchange, and are resilient to a shortage. Our proposed method like [40] can use cloud infrastructure for agents but also consider hierarchical regions(holons) with multiauthority and can plan for adaption on all levels together.

10 Conclusion

Due to the expanded use of the microgrids and management of them by independent private companies, we need proper control of them, which, due to the multi-authoring of the system, also is provided the possibility of proper stability. In this paper, a new architecture was presented for this purpose, which has a holonic and policy-based structure and by using the Internet communication infrastructure and appropriate communication protocols, it is possible to be stable in inappropriate conditions and policy can change during execution time. The components of control agents could develop different methods to perform the responsibilities of that component, therefore, considering that all the necessary infrastructure in DCAM architecture has been designed and simulated, it is a suitable platform for developing control of decentralized self-adaptive multi-authoring microgrids. The system source code is also available on Github.¹ This architecture makes it possible to use artificial intelligence and machine learning algorithms to properly predict and make appropriate decisions that would be done in future work.

Declarations

Conflict of interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Choi C, Esposito C, Wang H, Liu Z, Choi J (2018) Intelligent power equipment management based on distributed context-aware inference in smart cities. *IEEE Commun Mag* 18:212–217
2. Marzal S, Salas R, González-medina R, Garcerá G, Figueres E (2018) Current challenges and future trends in the field of communication architectures for microgrids. *Renew Sustain Energy Rev* 82:3610–3622
3. Frey S, Diaconescu A, Menga D, Demeure I (2015) A generic holonic control architecture for heterogeneous multiscale and multiobjective smart microgrids. *ACM Trans Auton Adapt Syst* 10(2):1–21
4. Khan MW, Wang J, Xiong L, Ma M (2018) Modelling and optimal management of distributed microgrid using multi- agent systems. *Sustain Cities Soc* 41(May):154–169

¹ <https://github.com/seyyed/DCAM>

5. Bui V, Member S, Hussain A, Member S, Kim H (2016) A multiagent-based hierarchical energy management strategy for multi-microgrids considering adjustable power and demand response. *IEEE Trans Smart Grid* 3053:1–10
6. Boudoudouh S, Maârrou M (2018) Multi agent system solution to microgrid implementation. *Sustain Cities Soc* 39:252–261
7. Ansari J, Gholami A, Kazemi A (2016) Multi-agent systems for reactive power control in smart grids. *Int J Electr Power Energy Syst* 83:411–425
8. Pahwa A et al (2015) Goal-based holonic multiagent system for operation of power distribution systems. *IEEE Trans Smart Grid* 6(5):2510–2518
9. Dehghanpour K, Nehrir H (2018) Real-time multiobjective microgrid power management using distributed optimization in an agent-based bargaining framework. *IEEE Trans Smart Grid* 9(6):6318–6327
10. Albaker A (2018) *Optimal Operation of Integrated Microgrids*. University of Denver
11. Akter MN, Mahmud MA, Haque ME, Oo AMT (2020) An optimal distributed energy management scheme for solving transactive energy sharing problems in residential microgrids. *Appl Energy* 270(April):115133
12. Khorasany M, Azuatalam D, Glasgow R, Liebman A, Razzaghi R (2020) Transactive energy market for energy management in microgrids: the Monash microgrid case study. *Energies* 13(8):1–23
13. Xu D et al (2020) Peer-to-peer multi-energy and communication resource trading for interconnected microgrids. *IEEE Trans Ind Inf* 3203:1–1
14. Kusakana K (2020) Optimal peer-to-peer energy management between grid-connected prosumers with battery storage and photovoltaic systems. *J. Energy Storage* 32:101717
15. Mohamed MA, Jin T, Su W (2020) Multi-agent energy management of smart islands using primal-dual method of multipliers. *Energy* 208:118306
16. Morstyn T, Member S, Hredzak B, Member S (2016) Control strategies for microgrids with distributed energy storage systems : an overview. *IEEE Trans Smart Grid* 3053:1–15
17. Ni J, Ai Q (2016) Economic power transaction using coalitional game strategy in micro-grids. *IET Gener Transm Distrib* 10(1):10–18
18. Yin S, Wang J, Qiu F (2019) Decentralized electricity market with transactive energy - A path forward. *Electr J* 32(4):7–13
19. Syptayev D, Kumar Nunna H S V S, Shintemirov A (2020) A novel peer-to-peer Negawatt trading transactive energy system for prosumers. in 2020 IEEE 14th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG) pp 181–186
20. Na U, Lee EK (2020) Fog BEMS: an agent-based hierarchical fog layer architecture for improving scalability in a building energy management system. *Sustainability* 12(7):1–28
21. Mishra S, Bordin C, Tomasgard A, Palu I (2019) A multi-agent system approach for optimal microgrid expansion planning under uncertainty. *Int J Electr Power Energy Syst* 109:696–709
22. Gomes L, Vale Z, Corchado JM (2020) Microgrid management system based on a multi-agent approach: an office building pilot. *Measurement* 154:107427D
23. Negeri E, Baken N, Popov M (2013) Holonic architecture of the smart grid. *Smart Grid Renew Energy* 4(2):202–212
24. Ansari J, Kazemi A, Gholami A (2015) Holonic structure: a state-of-the-art control architecture based on multi-agent systems for optimal reactive power dispatch in smart grids. *IET Gener Transm Distrib* 9(14):1922–1934
25. Dozein MG, Monsef H, Ansari J, Kazemi A (2016) An effective decentralized scheme to monitor and control the reactive power flow: a holonic-based strategy. *Int Trans Electr Energy Syst* 26(6):1184–1209
26. CEN-CENELEC-ETSI Smart Grid Coordination Group (2012) Smart grid reference architecture. <https://syc-se.iec.ch/deliveries/sgam-basics/>
27. Rajput PK, Sikka G (2021) Multi-agent architecture for fault recovery in self-healing systems. *J Ambient Intell Humaniz Comput* 12(2):2849–2866
28. Tom RJ, Sankaranarayanan S, Rodrigues JJPC (2020) Agent negotiation in an IoT-Fog based power distribution system for demand reduction. *Sustain Energy Technol Assess* 38:100653
29. Fanitabasi F, Pournaras E (2020) Appliance-Level Flexible Scheduling for Socio-Technical Smart Grid Optimization 8
30. Sampaio FC, Leão RPS, Sampaio RF, Melo LS, Barroso GC (2020) A multi-agent-based integrated self-healing and adaptive protection system for power distribution systems with distributed generation. *Electr Power Syst Res* 188(July):106525

31. Gollagi SG, Math MM, Daptardar AA (2020) A survey on pervasive computing over context-aware system. *CCF Trans Pervasive Comput Interact* 2(2):79–85
32. Di Marzo Serugendo G, Gleizes M-P, Karageorgos A (2005) Self-organization in multi-agent systems. *Knowl Eng Rev* 20(02):165–189
33. Bindelli S, Di Nitto E, Mirandola R, Tedesco R (2008) Building autonomic components: the SelfLets approach. *Automated software engineering - workshops, 2008. ASE Workshops 2008*:17–24
34. Kephart J, Chess D (2003) The vision of autonomic computing. *Computer* 36(1):41–50
35. Data.gov.au (2010) Smart-grid smart-city customer trial data - Dataset. <https://data.gov.au/dataset/ds-dga-4e21dea3-9b87-4610-94c7-15a8a77907ef/details>
36. Huang R, Xiao Y, Liu M, Shen X, Huang W, Peng Y, Shen J (2022) Multilevel dynamic master-slave control strategy for resilience enhancement of networked microgrids. *Energies* 15(10):3698
37. El Zerk A, Ouassaid M, Zidani Y (2022) Decentralised strategy for energy management of collaborative microgrids using multi-agent system. *IET Smart Grid* 5(6):440–462
38. Hamidi M, Raihani A, Youssfi M, Bouattane O (2022) A new modular nanogrid energy management system based on multi-agent architecture. *Int J Power Electron Drive Syst* 13(1):178–190
39. Jiménez VA, Lizondo DF, Araujo PB, Will AL (2022) A conceptual microgrid management framework based on adaptive and autonomous multi-agent systems. *J Comput Sci Technol* 22:1–11
40. Blechmann S, Sowa I, Schraven MH, Streblov R, Müller D, Monti A (2023) Open source platform application for smart building and smart grid controls. *Autom Constr* 145:104622

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.