# Interfacing Issues in Multiagent Simulation for Smart Grid Applications

IEEE Task Force on Interfacing Techniques for Simulation Tools

X. Wang, P. Zhang, Z. Wang, V. Dinavahi, G. Chang, J. A. Martinez, A. Davoudi, A. Mehrizi-Sani, and S. Abhyankar



***Abstract*—This paper discusses design and application of the multiagent simulation technology aiming to meet smart grid requirements. The difference between multiagent systems and multiagent simulation in smart grid applications is clarified. The state-of-the-art applications of multiagent simulation in power and energy systems are classified based on the simulation environment. The paper also addresses the interface issues including synchronization and data distribution for multiagent co-simulation. In addition, the emerging research paradigms in smart grid multiagent simulation are identified.**

***Index Terms*—Interface, multiagent simulation, multiagent systems (MAS), smart grid, time management.**

## I. Introduction

SMART GRID involves the use of information, communications, and distributed computing technologies to transmit and distribute electric energy more efficiently while improving the quality of power and system reliability. It allows the transmission and distribution system operators to integrate large-scale renewable and traditional generation resources or take precise actions to fix problems through alerts sent by smart sensors, controls, and distributed computing devices. With the aim of developing a smart grid, the grid is evolving from an entirely centralized structure to a decentralized one [1].

In the past decade, multiagent systems (MAS) technology has been recognized as a promising new paradigm for power grid planning, design and operation [2], [3]. MAS use a collection of heterogeneous and distributed intelligent agents which interact with each other and their environments to achieve specific outcomes. Agents exchange information with neighbors and with centralized controller if necessary, gather data from environment, and may be able to perform cognitive learning to adapt to changes in the system [4]. Therefore, MAS provide a common communication interface for all system components with autonomous control actions in a distributed and decentralized manner. It offers an effective way of serving as the platform for modeling autonomous decision-making entities and can be used to realize smart grid concepts for grid planning and operations.

There are typically two ways in which MAS can be applied in power systems. One is to employ MAS as a modeling approach to model physical systems and then determine inherent rules of the physical systems through simulations, such as the power marketplace simulation [5]–[7]. This multiagent modeling and simulation approach is also known as multiagent simulation (MASim) [8]. The other application is to utilize MAS as a method to manage physical systems that are composed of heterogeneous entities to achieve joint goals while considering the interests of each entity. Recently, several MAS-based applications have been reported in literature in the areas of load forecasting, fault location and diagnostics, protection design and training, etc. [9]–[15]. In these applications, one necessary step before implementing the designed MAS in reality is to verify the effectiveness and improve the performance of the utilized MAS approach in a simulated environment. Traditional power system simulation tools, which may work well for some specific purposes, are no longer able to provide holistic optimal solutions at this time [16]. MASim has to be used to set up agent models and simulate agent behaviors. However, MASim by its very nature lacks tools to model power systems and corresponding communication systems for its application to smart grids. As a result,

Manuscript received January 02, 2013; revised March 16, 2013; accepted April 24, 2013. Date of publication May 15, 2013; date of current version June 20, 2013. Paper no. TPWRD-00007-2013.

Task Force on Interfacing Techniques for Simulation Tools is with the Working Group on Modeling and Analysis of System Transients Using Digital Programs, IEEE Power and Energy Society, T&D Committee.

Task Force members: U. Annakkage, G. W. Chang, A. Davoudi, F. De Leon, V. Dinavahi (Task Force Chair), M. O. Faruque, S. Filizadeh, A. M. Gole, R. Iravani, J. Jatskevich, A. J. Keri, J. A. Martinez, A. Monti, L. Naredo, T. Noda, A. Ramirez, M. Rioual, V. Spitsa, M. Steurer, K. Strunz, X. Wang, P. Zhang, A. Mehrizi-Sani, and S. Abhyankar.

X. X. Wang is with the Carleton University, Carleton University, Ottawa, ON K1S 5B6 Canada.

P. Zhang is with the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269-2157 USA.

Z. Wang is with the School of Electrical Engineering, Shandong University, Jinan, Shandong 250061, China.

V. Dinavahi is with the Electrical and Computer Engineering Department, University of Alberta, Edmonton, AB T6G 2V4, Canada.

G. Chang is with the Electrical Engineering Department, National Chung Cheng University, Chia-Yi 621, Taiwan.

J. A. Martinez is with the Department Eng. Electrica-ETSEIB, Universitat Politecnica de Catalunya, Barcelona 08028, Spain.

A. Davoudi is with the Electrical and Computer Engineering Department, University of Texas, Arlington, TX 76011 USA.

A. Mehrizi-Sani is with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99164 USA.

S. Abhyankar is with the Argonne National Laboratory, Argonne, IL 60439 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRD.2013.2260566

MASim has to be interfaced with other simulation tools [16], [17].

MASim has been paid great attention as a decision support tool for predicting, understanding, and verifying hypotheses with the widespread use of the MAS technology in power systems. In this task force paper, an overview of MASim in smart grid applications is provided. Particular interest is given to the interfacing issues between the agent simulators, the available commercial off-the-shelf (COTS) simulators in power systems, and communication simulations. Future research topics, challenges, and opportunities in MASim are also identified.

The organization of this paper is as follows. The definitions and salient features of MASim are introduced in Section II. State-of-the-art applications of MASim in power systems and the architecture of MASim are summarized in Section III and Section IV, respectively. The interface issues of MASim are presented in Section V. The case studies of MASim are provided in Section VI and new research needs to meet the challenges in MASim are discussed in Section VII. This conclusion is in Section VIII.

## II. MultiAgent Simulation: Definition, Features, and Benefits

MASim is a modeling concept based on MAS which are systems that comprise of two or more intelligent agents [2], [3]. An agent is defined as "a software (or hardware) entity that is situated in some environment and is able to autonomously react to changes in that environment" [18]. The environment is a shared structured entity for communication between agents.

MASim is used to simulate a real system of multiagent nature, involving many different components and interactions amongst agents in a complex, diverse way. In MASim, one extensively details the decision processes of simulated agents at a micro-level. At the macro-level, structures are described as a result of interaction of agents at the micro-level. The basis of MASim is the interaction among agents as a central point.

Agents are autonomous entities that react to environmental change and proactively change their behavior [8]. This feature would allow MASim, which is used to simulate MAS, to test if the smart grid is able to heal itself, provide higher quality power that will save money wasted from outages, accommodate all generation and storage options, enable electricity markets to flourish, run more efficiently, and enable higher penetration of intermittent power generation sources [20]. The agents' action and perception are local, though having flexible interaction between agents in order to perform these smart grid functions. MASim has variable structure models that entail in their description the possibility to change their own structure; their constitutive components as well as the relations that exist among them [21]. This feature is what makes MASim adapt well to the smart grid. The advantages and disadvantages of MASim are summarized as follows [8].

### A. Advantages of MASim

1) can deal with MAS directly (real agent versus simulated agent);
2) facilitates structural validation;

TABLE I
Summary of the MASim Applications in Power Engineering

| MASim Application | MAE | MAF | HIL |
|---|---|---|---|
| Load forecast | [23] | [24] | |
| Optimal power flow | | [25] | |
| Protection design and training | | [26]-[28] | |
| Generation expansion planning | [29]-[32] | | |
| Fault location and diagnostics | [33] | [34][35] | |
| EMS and control center applications | [36]-[39] | | [40] |
| Online transient stability analysis and control | | [41]-[43] | |
| Electricity market simulators | [44] | [45] | |
| Microgrid control and scheduling | | [46]-[54] | [55]-[57] |

3) elegant treatment of variable structures;
4) enable model adaptation and evolution;
5) easy to model heterogeneous space and population;
6) provides different levels of observation.

### B. Disadvantages of MASim

1) development of complex models can be expensive;
2) difficult to determine minimal model;
3) established formalism is missing, difficult to document;
4) calibration problem; finding the best parameter setting for a model (given a structurally valid model);
5) sensitivity problem; even small changes may have a large effect;
6) limited comparison done with centralized coordination.

## III. Applications of MASim in Power and Energy Systems

If MASim is required for power engineering applications, one may develop user-specific power component models in the MASim environment. However, this is a very time consuming process and it is infeasible to some degree for large-scale simulations. A viable method is to select an appropriate COTS simulation tool and integrate it with multiagent simulators. This approach can build an integrated simulation system, that is, a multiagent co-simulation (MACSim) system. In this kind of co-simulation, multiagent simulators are responsible for modeling and simulating of agents and the COTS simulators are responsible for modeling and simulating other professional domains. In MACSim, agents can be used as a mechanism for combining simulators [22] or they can be used to represent key entities such as protective relays [16].

State-of-the-art applications of MASim in power and energy systems are summarized in this section. Three categories of MASim are identified according to the interconnectivity between the MASim execution environment and the external simulation environment. The categories are denoted as multiagent environment (MAE) simulation, multiagent federation (MAF) simulation, and hardware-in-the-loop (HIL) simulation, respectively. Table I lists the summary of the MASim applications in these categories.

The specifics of the three simulation approaches are described as follows.

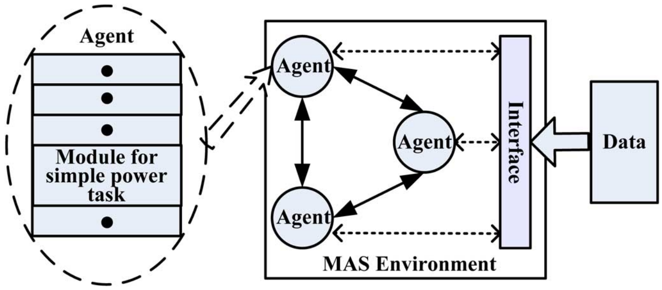


Fig. 1. Framework of the MASim implemented only in the MAS environment.

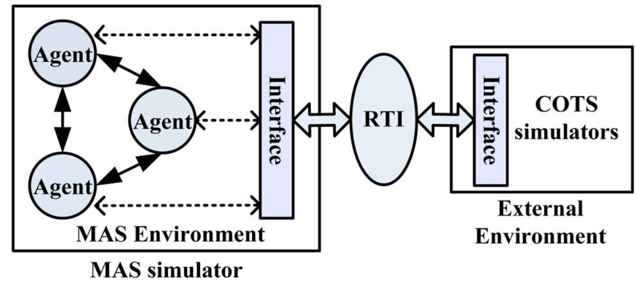


Fig. 2. Framework of the federated MASim.



### A. *Multiagent Environment (MAE) Simulation*

Most agent platforms are capable of both developing MAS and providing environments where MAS can be implemented. Multiagent missions can be simulated only in the MAS environment if they do not need to interact with external environment in real time [44]. The framework of such a simulation is shown in Fig. 1. Special modules need to be developed simultaneously with the progress of MAS development, if necessary, to deal with tasks concerned with power computation or simulation, in which case MASim does not need to resort to COTS simulation systems [29], [36]–[39]. Another kind of special module is the data interface which is in charge of managing non-real-time data input/output (I/O). There are two sources for data. The first source is the database of the real power system, such as the supervisory control and data acquisition (SCADA) system and digital fault recorder [23].

Another source is the data that has already been produced by the utilized COTS power simulators which do not interact with MASim and then do not need to develop interface specially designed in them [33]. MASims adopting these methods are commonly used to simulate ex post facto MAS missions, such as fault location MAS applications [33].

### B. *Multiagent Federated (MAF) Simulation*

In most cases, MAS environments need to interact in real-time with external environments which can be provided by COTS software systems such as power simulators and communication simulators, to fulfill their simulation missions. Three types of service that can be provided by COTS software systems were discovered:

*1) Providing Virtual Object Environments:* Some MAS missions (e.g., control applications) must interact with the actual power systems. Therefore it is necessary to employ COTS power simulators to establish a virtual object environment for such MASims. An example can be found in [45]–[47], [50]–[52]. The most important point one should recognize is that the object environment must run synchronously with the MAS environment.

*2) Providing Calling Services for Professional Computation or Simulation:* In some other MASims, because MAS have no capability to complete certain professional computing or simulating tasks, it is necessary to resort to external COTS software systems to fulfill the tasks. The COTS software systems can be considered as special modules called by MASim. For example, in [49] a micro grid PoolCo market for distributed energy resource (DER) management is simulated based on MAS. The MASim employs Power World Simulator to check the reliability and to mitigate the congestion of the investigated microgrid.

*3) Providing Communication Simulation Environments:* Distributed application is one of the most important characteristics of MAS. Only through computer networks agents of MAS can interact with each other. For some time-sensitive MAS missions, it is necessary to investigate the effect of network latency through COTS communication simulators which are also commonly utilized to evaluate communication protocols. For example, Network Simulator 2 (NS2), a discrete-event-driven communication network simulator, is adopted by [16] to evaluate new communication protocols.

In MAF simulations, it is critically important to create a runtime infrastructure (RTI) [57] to glue the MAS environment and the external environment composed of COTS simulators. Each simulator is referred to as a federate. The collection of federates interacting through the RTI is referred to as a federation [59]. The framework for this type of MASim is shown in Fig. 2 where the role of RTI is to route all messages between the component simulators of MASim and ensure that the simulation clocks are synchronized across all the simulators. It is clear that interfaces are necessary both in the MAS execution environment and in the external environment. These interfaces should be able to provide mutual data exchanging services for the MAS environment and the COTS simulators.

### C. *Hardware-in-the-Loop (HIL) Simulation*

HIL simulation is a real-time simulation with the hardware components participating in the simulation loop. The HIL technique has also been applied in MASim because of the advantages of test environment, build-up time and development cost for developing and testing an embedded system [60]. The hardware systems employed by MASim usually have the following functions:

*1) Providing Hardware Object Environments:* Test-bed composed of hardware provides a real-time and more detailed object environment for MASim because COTS software power simulators are not capable of modeling all details of real power systems. Reference [55] set up a laboratory-based power system infrastructure for interconnection issue studies of micro grid and distributed generation. A MAS framework was designed and tested in this HIL platform.

*2) Providing Hardware Communication Evaluation Environments:* Hardware simulation is a kind of evaluation to study

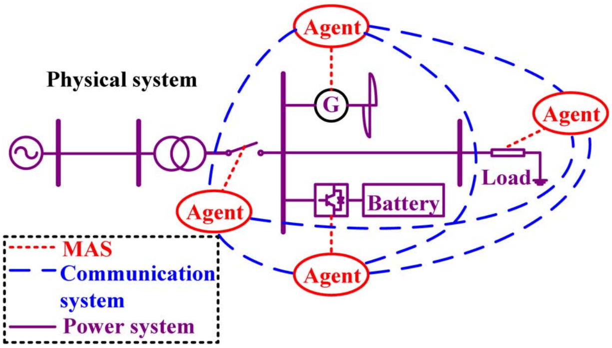


Fig. 3. Example physical system for MACSim.

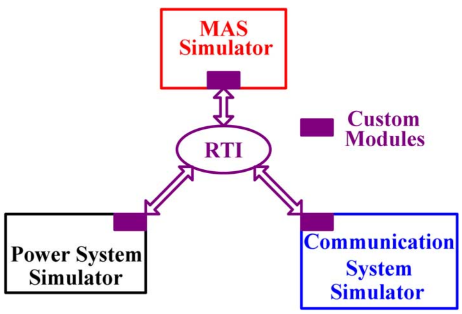


Fig. 4. Architecture of MACSim.

communication issues such as time delays and protocols for MAS. Evaluating communication performance of MAS through network hardware is more convincing. Various communication scenarios were tested in [56] for both intraplatform and interplatform communications.

## IV. Architecture of Multiagent Co-Simulation

As indicated in Section III, MACSim which combines different component simulators is the representative MASim method used in smart grid applications. Fig. 3 shows an example system for MACSim application. The example system is composed of three physical subsystems: the power system (PS), the communication system (CS), and the MAS. Each subsystem can be simulated by the specialized COTS simulator individually. Fig. 4 illustrates the integrated simulation system for the physical system shown in Fig. 3. The simulation system consists of three specialized COTS simulators and an RTI.

In Fig. 4, the MAS simulator performs MASim; the CS simulator provides communication simulation services under the request of the MAS simulator; and the PS simulator provides power engineering simulation services under the request of the MAS simulator. It is important to choose the right simulators when creating such a co-simulation system. Typical MAS simulators include JADE, FIPA-OS, ZEUS, JACK, etc. [61], and CS simulators include NS2, OPNET, J-Sim, OMNet++, etc. [62].

For each individual simulator, custom modules must be developed as interfaces to be integrated into the co-simulation system. The RTI is a distributed operating system that provides a set of services used to interconnect the simulators. The services include federation, declaration, object, ownership, and time and data distribution managements [59]. Among these services, time management (TM) and data distribution management (DDM) are particularly concerned in this paper and the related interface issues will be discussed in details in Section V.

## V. Interfacing Issues in MACSim

MACSim is based on federated simulation with continuous time (PS) and discrete-event (CS) simulation tools [63]. MAS simulator is also based on a discrete time model where system states are changed at discrete time points only [64]. In MACSim, there are some key issues to be solved in order to make the co-simulations work well. Ensuring the synchronization of the simulation clocks between all simulators is a challenge during the development of federated MASim systems. Designing interface module and data exchange channel is also essential. Typically, high-level architecture (HLA) is used to handle these interfacing issues in distributed computer simulations [65], [66]. HLA is defined by three components: a set of rules for distributed simulations, an interface specification for RTI, and a standard object model template describing the information communicating between the simulations.

### A. Time Management

MACSim synchronization is realized by TM which not only guarantees that events are processed in the correct order but also ensures that the repeated executions of a simulation with the same inputs produce exactly the same results [59]. TM is typically supported by TM mechanisms built in RTI [67]. TM mechanisms can be classified into time-stepped mechanism and event-driven mechanism. Time-stepped mechanism advances simulation in equidistant time steps. The component simulator does not proceed to the next time step until all simulation activities associated with the current time step have been completed. Event-driven mechanism processes system events in time stamp order. The system state is changed only if an event is triggered in this mechanism [59].

An event-driven mechanism may be known as conservative, when the mechanism introduces constraints to avoid causality errors (out of time or logical order messages), or optimistic, when the mechanism allows causality errors and provides suitable techniques to recover from an incorrect system state [64].

TM mechanisms of MACSim can be better explained by the following example where a multiagent-based protection system [17] is co-simulated through an agent simulator developed in JAVA environment (JADE) and an electromagnetic transient simulator, PSCAD/EMTDC. The communication process is implemented in JADE in [17], whereas a communication module is assumed to be built to simulate the physical communication system in this example to clarify the time delay issue in MASim. Fig. 5 shows the protection scheme consisting of two relays placed at the end of a transmission line. Each relay is represented by an agent (R1 and R2) in the agent simulator. When a fault (high current) occurs on the line, the relays will both see the fault and then R1 communicates with R2 by sending a message to ask whether R2 saw the fault. If so, the fault is judged to be located within the protection zone of the relays and the circuit breakers CB1 and CB2 will be opened by R1 and R2 to isolate the fault.

There are two types of uncontrollable time delay (UTD) in the MACSim shown in Fig. 5: simulation time uncontrollable time

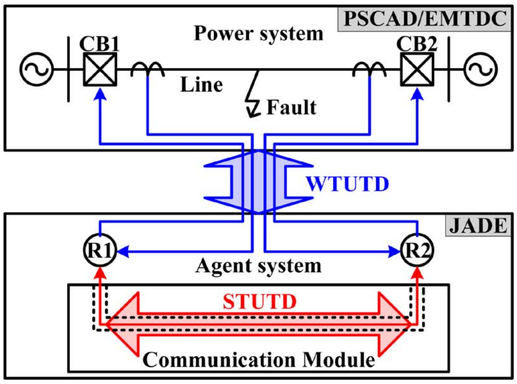


Fig. 5. Transmission line with pilot protection [17].

delay (STUTD) and wall clock time uncontrollable time delay (WTUTD). Simulation time (ST) refers to simulator's representation of physical time (PT) and wall clock time (WT) is the time when simulator is executed [59]. Correspondingly, STUTD is the time delay caused by simulation of physical systems (e.g., 1s time delay could be generated by the communication module when simulating the communication system in Fig. 5). This time delay is actually determined by simulation condition settings (e.g., different communication methods) and it could be controlled by users. WTUTD is the time delay caused by different simulators during data exchange or synchronization processes. This time delay is determined by simulation tools themselves, however it will not affect simulation results if appropriated TM mechanisms are designed. More detailed discussions on time delay of MAS can be referred to [10] where rules are recommended for dealing with communication latency.

Fig. 6 displays the simulation process of the illustrated system if no TM mechanism is employed in the MASim. In Fig. 6, the PT axis is represented by the unit time (ut) with the base value of the preset time step. It is important to note that in order to simplify the analysis in this illustration, ST is advanced to be synchronized with WT. In other words, it is assumed that the co-simulation is a "virtual" real-time simulation.

In the simulation scheme, PSCAD/EMTDC produces a fault at the instant 4 ut. The fault information is then delivered to R1 and R2 in the agent simulator. This data exchange process is C1 which consumes the time segment of 8 ut. During this process, the simulation executed by PSCAD/EMTDC does not pause, but continues to advance. At 12 ut, the relay agents receive the fault information. Agent R1 sends a request-response-type message to agent R2, telling R2 that R1 monitored a high current phenomenon and requesting a response from R2. This process is A2 in the agent simulation. It is assumed that the time delay of the single communication between agents is set as 1 ut by the communication module and the time consumption of agents' action is neglected. So A2 can be considered as a communication process taking 1 ut.

At 13 ut, R2 receives the message from R1. It should be noticed that R2 has received the fault information at 12 ut from PSCAD/EMTDC. So R2 sends an order of opening CB2 through the data exchange process C2 at 13 ut. Meanwhile, R2 sends a response to R1, telling R1 that R2 also monitored a high

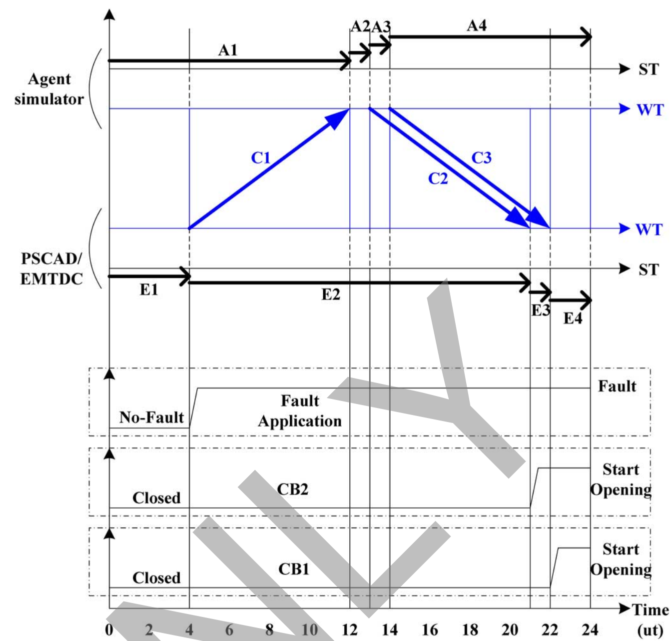


Fig. 6. Simulation process of the protection system without TM mechanism.

current phenomenon. The response communication process A3 also consumes 1 ut. At 14 ut, R1 receives the response from R2 and then sends an order of opening CB1 through the data exchange process C3. Because the order sent from R1 is 1 ut later than that from R2, the starting opening operation of CB1 should also be 1 ut later than that of CB2. The simulation result produced by PSCAD/EMTDC verifies the aforementioned conclusion, that is, CB2 starts opening at 21 ut and CB1 starts opening at 22 ut.

From the aforementioned analysis, it is clear that the time delays between the fault and the circuit breaker operations reflect mostly the data exchange time delays. This is because the co-simulation system has no TM mechanism. The absence of TM leads to several consequences. The most serious one is that the simulation results do not meet the standard on the repeatability of results. In other words, the incontrollable time delays of the data exchange process between the agent simulator and the PS simulator will cause different simulation results. For instance, if the time delay of data exchange is not 8 ut but 6 ut in the example, the starting opening of CB2 will move from 21 to 17 ut, assuming the other conditions of co-simulation do not change.

In order to deal with this problem, a TM mechanism must be added to the co-simulation system. The following part presents the effect of different TM mechanisms on the co-simulation.

*Time-Stepped Mechanism:* Fig. 7 reveals the simulation process of the example system with the time-stepped TM mechanism. In this mechanism, each component simulator executes independently until all simulators' ST reaches a synchronization point where the simulators are able to interact with each other by sending messages through data exchange processes. Once the simulator interactions have ended, all of the simulators continue to run until the next synchronization point is reached. In this case, the time segment of 1 ut is set

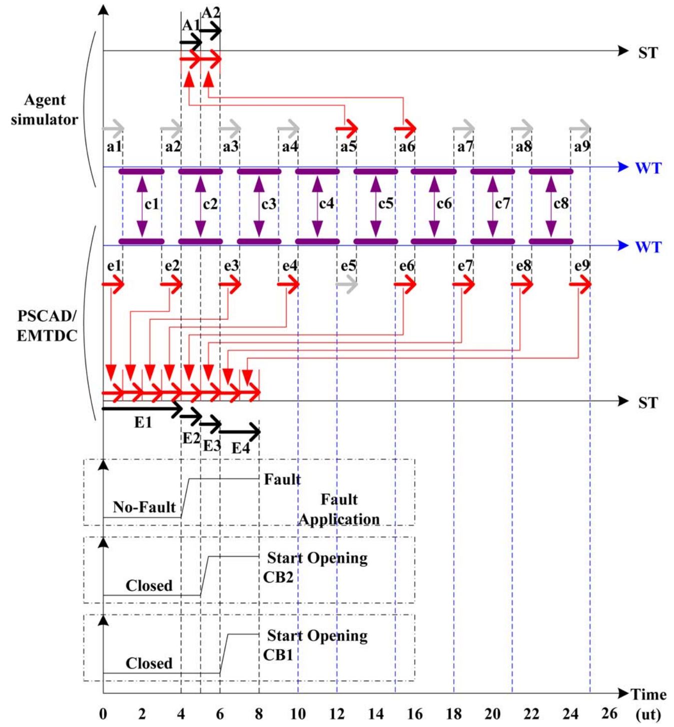


Fig. 7. Simulation process of the protection system with the time-stepped TM mechanism.

between two adjacent synchronization points and the time delay of data exchange processes is assumed as 2 ut.

In Fig. 7, on the WT axis of the component simulators, c1–c8 are the periods for bidirectional data exchange processes; a1–a9 represent the execution periods between two neighboring synchronization points in the agent simulator; and e1–e9 represent the execution periods between two consecutive synchronization points in PSCAD/EMTDC. E1–E4 are the simulation periods on the ST axis when system states are changed in PSCAD/EMTDC. A1–A2 are corresponding to the similar periods in the agent simulator.

During the periods of c1–c8, the agent simulator and PSCAD/EMTDC halt their simulation on the ST axis. Due to this reason, the advancement of the WT axis is different from that of the ST axis in this case. According to the simulation setting, PSCAD/EMTDC runs through e1–e4 on the WT axis or E1 on the ST axis before the fault occurs. When the fault happens at 10 ut on the WT axis or 4 ut on the ST axis, the fault information with simulation time stamp (STS) is sent to R1 and R2 via c4. PSCAD/EMTDC stops to advance at this time. Through c4, the relay agents know that there was fault in the power system at 4 ut (ST) from the fault information with STS, then the agent simulation starts from 4 ut (ST). Because the agent simulator needs time to execute simulation and make decisions, it is impossible for the agents to send executable orders back to PSCAD/EMTDC through c4 before 12 ut (WT). Since PSCAD/EMTDC has not received executable orders from the agents, it does not advance the power system simulation until the next synchronization point. In other words, e5 is a null execution in PSCAD/EMTDC. Similarly, a1–a4 and a7–a9 of the agent simulator are also null executions.

At 12 ut (WT), the agent simulator starts to perform the execution of a5, in which R1 sends a request-response-type message to R2 informing R2 that R1 observed a high current phenomenon near CB1 and asking R2 if it observed a high current phenomenon. After 1 ut of communication time delay, R2 receives the message from R1, and sends a response telling R1 that R2 also observed a fault. Meanwhile, R2 makes an order to open CB2. At this time, that is, 13 ut (WT), the agent simulation is just advanced to the synchronization point, so the agent simulator has to stop and starts to exchange data with PSCAD/EMTDC through c5. The order made by R2 will be received by PSCAD/EMTDC through c5. Since no useful information is sent from PSCAD/EMTDC after c5, the agent simulator continues to run into a6. After 1 ut of communication time delay, R1 receives the response from R2 and then makes an order to open CB1 at the end of a6. Then the order made by R1 will be sent to PSCAD/EMTDC through c6. In this agent simulation, a5 and a6 on the WT axis are mapped into A1 and A2 on the ST axis, respectively.

At 15 ut (WT), PSCAD/EMTDC starts to advance the power system simulation again because it received the order of opening CB2 at this time. Remember PSCAD/EMTDC is halted at 4 ut (ST) before, so the new advancement will go from 4 ut (ST) to 5 ut (ST). This period e6 on the WT axis is corresponding to E2 on the ST axis. At the end of E2, the starting opening of CB2 is simulated. The result can also be seen from the bottom part of Fig. 7 where CB2 starts opening after 5 ut on the time axis. Before 18 ut (WT), PSCAD/EMTDC received the order to open CB1 at 6 ut (ST) via c6, and the co-simulation system commands that all simulators must stop at 8 ut (ST). Therefore, the power system simulation is advanced though e7–e9 and then stops. At the end of e7, which is corresponding to E3 on the ST axis, CB1 begins opening.

From the aforementioned analysis, one can see that the simulation processes are logically synchronized on the ST axis and the circuit breaker operations will always happen at 5 ut and 6 ut on the ST axis no matter how long the data exchange processes and the communication delays between JADE and PSCAD/EMTDC are. Thus, the time-stepped TM mechanism is effective to solve the synchronization problem for MACSim [16]. However, this mechanism can introduce accumulating time errors due to difficulty for selecting a perfect synchronization boundary [63]. This shortcoming of the time-stepped mechanism can be overcome by the event-driven mechanisms.

*Event-Driven Conservative Mechanism:* The illustration of the event-driven conservative mechanism is shown in Fig. 8. The difference between the event-driven conservative mechanism and the time-stepped mechanism is that the component simulators do not need to stop for every synchronization point in the former mechanism. In other words, the advancement of the simulators is triggered by the events in the event-driven mechanism (e.g., messages sent from the agents to PSCAD/EMTDC).

When a fault occurs at 4 ut (ST), PSCAD/EMTDC is paused until it receives orders from the agent simulator. The fault information is delivered to R1 and R2 at the same time. At 12 ut (WT), the agents receive the related messages with STS from the PS simulator and then the agent simulator starts to advance at 4 ut (ST) which is known from STS.

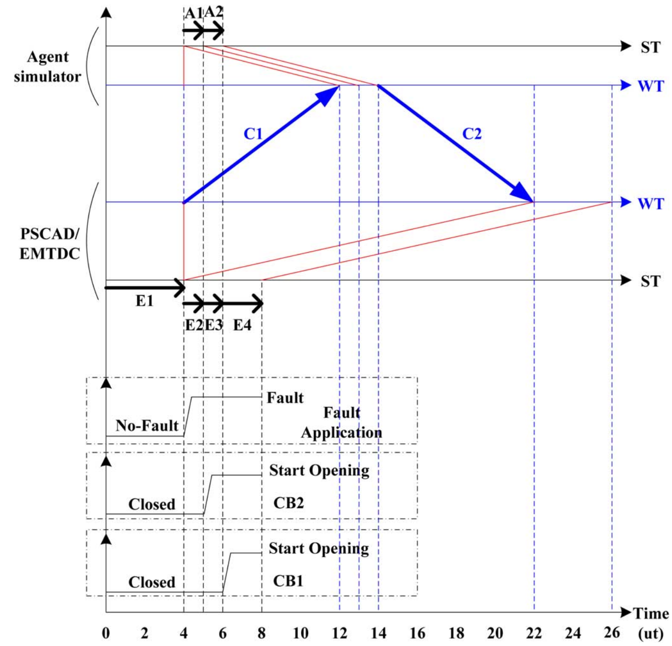


Fig. 8. Simulation process of the protection system with the event-driven conservative mechanism.

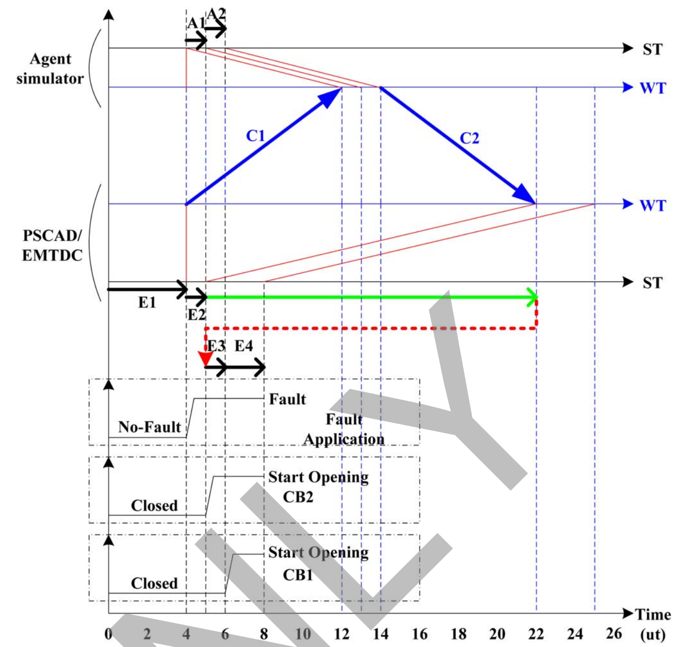


Fig. 9. Simulation process of the protection system with the event-driven optimistic mechanism.

The following logic processes (LPs) in this mechanism are similar with those in the time-stepped mechanism until R2 sends an order to PSCAD/EMTDC to open CB2 at 5 ut (ST). This order is not directly sent to PSCAD/EMTDC but saved temporarily in the agent simulator. After 1 ut (ST), R1 receives the response from R2 and sends its decision to PSCAD/EMTDC that CB1 should start opening at 6 ut (ST).

The second order is also saved in the agent simulator. The agent simulator sends the two orders together to PSCAD/EMTDC at 13 ut (WT) after judging that no more orders will be made by the agents. Because the data exchange process C2 has the time delay of 8 ut (WT), PSCAD/EMTDC receives the first order from the agent simulator at 22 ut (WT). The circuit breakers will start opening at 5 ut (ST) and 6 ut (ST) eventually.

*Event-Driven Optimistic Mechanism:* The event-driven conservative mechanism avoids causality violation by sacrificing the efficiency of co-simulations. The event-driven optimistic mechanism shown in Fig. 9, however, can provide a more efficient service.

In Fig. 9, PSCAD/EMTDC continues to run after the fault information is delivered to related agents at 4 ut (ST). At 22 ut (WT), PSCAD/EMTDC receives the orders from JADE that CB2 and CB1 should open at 5 ut (ST) and at 6 ut (ST), respectively. But the PS simulator has already advanced to 22 ut (ST) and the starting of the circuit breaker opening was not simulated because PSCAD/EMTDC did not receive the orders at 5 ut (ST) or 6 ut (ST). So the power system simulation must roll back to the state of 5 ut (ST) and, hereafter, re-simulate the opening of CB2 and CB1.

At 4 ut (ST), PSCAD/EMTDC did not stop the power system simulation, but "optimistically" assumed that continuing to advance the simulation would not violate the causality constraints. When the system observed the violation at 22 ut (WT), a rollback mechanism was employed to recover the correctness of the simulation.

It should be noted that the event-driven optimistic mechanism consumes 1 ut (WT) less than the event-driven conservative mechanism. The reason is that PSCAD/EMTDC has already executed E2 before it received the orders from JADE. This action increases the simulation efficiency.

### *B. Interface Module Development*

For the MACSim architecture shown in Fig. 4, each component simulator or federate needs to have an interface module to communicate with RTI. A two-part interface based on the ambassador paradigm was defined in HLA [68]. In this paradigm, a federate communicates with a RTI using its RTI ambassador. Conversely, a RTI communicates with a federate via federate ambassador [58]. Four interface implementation methods for a simulator to comply with an RTI are listed in [58].

1) Re-implementation of the tool with HLA-extensions: In this method, the source code of the simulator is modified to interface with the RTI if the code can be accessed.
2) Extension of intermediate code: If the simulator can generate intermediate source code in a higher level language, the intermediate code can be programmed to realize HLA extensions.
3) Usage of an external programming interface: This method utilizes the library interface (e.g., dynamic link library interface) with the function call ability to support RTI.
4) Coupling via a gateway program: If the previous three methods do not work for a simulator, a gateway program can be developed for the simulator to communicate with RTI through files, pipes or sockets.

In the MACSim system presented in [16], NS2 uses the second method to implement the interface module for RTI, PSCAD/EMTDC employs the third method, and PSLF applies the fourth method.

### C. Data-Exchange Channel

After the establishment of costumer interface modules in component simulators, data exchange channels should also be built between RTI and simulators. The development of data exchange channels is related to inter-process communication (IPC) which is a set of methods for data exchange among multiple threads in one or more computing processes [69]. Socket communication is one of the IPC methods which have been applied in MACSim. In [48], the network socket communication based on TCP/IP protocols is used to exchange data between a micro grid simulated in Matlab/Simulink in one computer and a MAS simulator created in JADE in another computer. The server/client structure is exploited by the TCP/IP model. Similar communication method is also used in [46] where an agent-based control system developed in JADE communicates with a hybrid micro grid set up in RTDS. Reference [17] utilizes a local socket communication method to realize IPC between an agent simulator developed in Java and a transmission power system simulated in PSCAD/EMTDC in one computer.

Some MACSim systems use file and shared memory for fast data exchange when the component simulators are at the same physical location. For example, files are utilized to exchange information between RTI and PSLF in [16]. Some COTS tools are also available for MACSim IPC such as EZJCom [50] and the IBM development tool for Java-COM Bridge [49].

## VI. Emerging Research Paradigms

In smart grid applications, MACSim systems can be classified into the following three categories according to their advance and maturity.

*Case-by-Case:* This kind of MACSim systems is developed for a special case application. The simulation system must reconfigure its time management or data distribution management if another case is going to be simulated in the system.

*Platform:* The MACSim systems in this category commonly consist of fixed individual simulators and have special applications. The users no longer involve with integrating issues such as time management and data distribution management and only focus on application.

*Framework:* In this category, the MACSim systems have core RTI software and a series of interface standards. The RTI software is responsible to provide RTI services such as time management and data distribution management. Any COTS simulators that match its interface standards are capable of integrating with the RTI software. For some COTS simulators, users can develop interface modules to match these interface standards. For a simulation application, users can select special COTS simulator to compose a MACSim system and then perform the simulation in it. One of the most significant advantages of this kind of MACSim systems is that users can select appropriate COTS simulators to integrate with RTI software and then perform simulations.

Until now, most of the existing MACSim systems are case-by-case. Only few of them are platform based, such EPOCHS [9]. There is still work to be done for framework MACSim systems. Some directions for development of MACSim may include: 1) new design philosophy and architecture; 2) efficient and intelligent communication manager for agents coordination, such as transparent but secure and energy efficient communication protocols, uniform agent communication languages and a scalable architecture extensible for future applications; 3) new algorithms for more efficient time management and data distribution management; and 4) new criteria or standards for MAS-oriented interfacing, which helps to develop framework MACSim.

## VII. Conclusion

This task force paper presents the state-of-the-art development of MASim for smart grid requirements. MASim provides decision support tools to evaluate the effectiveness of MAS in the decentralized smart grid applications. Agent simulators together with power system simulators and communication simulators are necessarily co-simulated to fulfill the objectives of MASim. The efficient federation of these COTS simulators must carefully deal with the interfacing issues such as synchronization and data distribution. It is recognized that emerging research efforts are needed in design and implementation of the MASim framework in power engineering fields.

## References

[1] R. Roche, B. Blunier, A. Hialire, and A. Koukam, "Multi-agent systems for grid energy management: A short review," in *Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc.*, 2010, pp. 3341–3346.

[2] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications—Part I: Concepts, approaches, technical challenges," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1743–1752, Nov. 2007.

[3] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications—Part II: Technologies, standards, tools for building multi-agent systems," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1753–1759, Nov. 2007.

[4] M. Luck, P. McBurney, and C. Preist., Agent technology: Enabling next generation computing: A roadmap for agent-based computing. Southampton, U.K., 2003. [Online]. Available: http://www.agentlink.org/roadmap/al2/roadmap.pdf, AgentLink Rep.

[5] P. Wei, Y. Yan, Y. Ni, Y. Jerome, and F. F. Wu, "A decentralized approach for optimal wholesale cross-border trade planning using multi-agent technology," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 833–838, Nov. 2001.

[6] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4603–4612, Oct. 2011.

[7] A. J. Bagnall and G. D. Smith, "A multiagent model of the UK market in electricity generation," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 522–536, Oct. 2005.

[8] F. Pecora., "Multi-agent simulation: Principles of simulation and the multi-agent approach.," Lecture Notes. Örebro, Sweden, 2011. [Online]. Available: http://aass.oru.se/~fpa/teaching/MAS-lecture-1.pdf

[9] A. A. Aquino-Lugo, R. Klump, and T. J. Overbye, "A control framework for the smart grid for voltage support using agent-based technologies," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 173–180, Mar. 2011.

[10] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 57–64, Jun. 2010.

[11] K. M. Rogers, R. Klump, H. Khurana, A. A. Aquino-Lugo, and T. J. Overbye, "An authenticated control framework for distributed voltage support on the smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 40–47, Jun. 2010.

[12] Y. Xu and W. Liu, "Novel multiagent based load restoration algorithm for micro grids.," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 152–161, Mar. 2011.

[13] J. F. Borowski, K. M. Hopkinson, J. W. Humphries, and B. J. Borghetti, "Reputation-based trust for a cooperative agent-based backup protection scheme," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 287–301, Jun. 2011.

[14] P. H. Nguyen, W. L. Kling, and P. F. Ribeiro, "Smart power router: A flexible agent-based converter interface in active distribution networks," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 487–495, Sep. 2011.

[15] J. Liu, A. Benigni, D. Obradovic, S. Hirche, and A. Monti, "State estimation and branch current learning using independent local Kalman filter with virtual disturbance model," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 9, pp. 3026–3034, Sep. 2011.

[16] K. Hopkinson, X. Wang, R. Givanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: A platform for agent-based electric power and communication simulations built from commercial off-the-shelf components," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 548–558, May 2006.

[17] M. Baran, R. Sreenath, and N. R. Mahajan, "Extending PSCAD/EMTDC for simulating agent-based distributed applications," *IEEE Power Eng. Rev.*, vol. 22, no. 12, pp. 52–54, Dec. 2002.

[18] M. Wooldridge and G. Weiss, "Intelligent agents," in *Multi-Agent Systems*. Cambridge, MA: MIT Press, Apr. 1999, pp. 3–51.

[19] D. Weyns, H. Parunak, F. Michel, T. Holvoet, and J. Ferber, "Environments for multiagent systems: State-of-the-art and research challenges," in *Proc. E4MAS*, 2004, pp. 1–47.

[20] A. Rohit, Smart grids for energy distribution, infrastructure and efficiency. Mar. 2007. [Online]. Available: http://www.eco20-20.com/smart-grids-for-energy-distribution-infrastructure-and-efficiency.html

[21] A. M. Uhrmacher and B. P. Zeigler, "Variable structure models in object-oriented simulation," *Int. J. General Syst.*, vol. 24, no. 4, pp. 359–375, 1996.

[22] L. F. Wilson, D. Burroughs, J. Sucharitaves, and A. Kumar, "An agent-based framework for linking distributed simulations," in *Proc. Winter Simulation Conf.*, Orlando, FL, 2000, vol. 2, pp. 1713–1721.

[23] R. Tsai and J. L. Chen, "Design of a distributed problem-solving system for short-term load forecasting," in *Proc. 35th Midwest Symp. Circuits Syst.*, 1992, vol. 1, pp. 395–399.

[24] P. H. Nguyen, W. L. Kling, G. Georgiadis, M. Papatriantafilou, L. A. Tuan, and L. Bertling, "Distributed routing algorithms to manage power flow in agent-based active distribution network," in *Proc. Innovative Smart Grid Technol. Conf. Eur.*, Gothenburg, Sweden, 2010, pp. 1–6.

[25] P. H. Nguyen, W. L. Kling, and J. M. A. Myrzik, "Power flow management in active network," in *Proc. IEEE Bucharest Power Tech. Conf.*, Bucharest, Romania, 2009, pp. 1–6.

[26] R. Giovanini, K. Hopkinson, D. V. Coury, and J. S. Thorp, "A primary and backup cooperative protection system based on wide area agents," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1222–1230, Jul. 2006.

[27] Z. He, Z. Zhang, X. Yin, Z. Li, X. Kong, and Y. Wang, "Multi-agent based wide area backup protection for series compensated lines," in *Proc. Asia-Pacific Power Energy Eng. Conf.*, Wuhan, China, 2011, pp. 1–4.

[28] C. P. Nguyen and A. J. Flueck, "Agent based restoration with distributed energy storage support in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 1029–1038, Jun. 2012.

[29] J. Yen, Y. H. Yan, B. J. Wang, P. K. H. Sin, and F. F. Wu, "Multi-agent coalition formation in power transmission planning," in *Proc. 31st Hawaii Int. Conf. System Sci.*, Kohala Coast, HI, 1998, vol. 4, pp. 433–443.

[30] M. A. Ortega-Vazquez and D. S. Kirschen, "Assessment of generation expansion mechanisms using multi-agent systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting—Convers. Del. Elect. Energy 21st Century*, Pittsburgh, PA, 2008, pp. 1–7.

[31] A. Botterud, M. R. Mahalik, G. Conzelmann, R. F. Silva, S. Vilela, and R. Pereira, "Multi-agent simulation of generation capacity expansion decisions," in *Proc. EEM 5th Int. Conf. Eur. Elect. Market*, Lisboa, Portugal, 2008, pp. 1–6.

[32] A. Botterud, M. R. Mahalik, T. D. Veselka, H. S. Ryu, and K. W. Sohn, "Multi-agent simulation of generation expansion in electricity markets," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Tampa, FL, 2007, pp. 1–8.

[33] K. Nareshkumar, M. A. Choudhry, J. Lai, and A. Feliachi, "Application of multi-agents for fault detection and reconfiguration of power distribution systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Calgary, AB, Canada, 2009, pp. 1–8.

[34] X. Zeng, K. K. Li, W. L. Chan, and S. Su, "Multi-agents based protection for distributed generation systems," in *Proc. IEEE Int. Conf. Elect. Utility Dereg., Restruct. Power Technol.*, 2004, pp. 393–397.

[35] G. Zhabelova and V. Vyatkin, "Multiagent smart grid automation architecture based on IEC 61850/61499 intelligent logical nodes," *IEEE Trans. Ind. Electron.*, vol. 59, no. 5, pp. 2351–2362, May 2012.

[36] M. Mao, W. Dong, and L. Chang, "Design of A novel simulation platform for the EMS-MG based on MAS," in *Proc. IEEE Energy Convers. Congr. Expo.*, Phoenix, AZ, 2011, pp. 2670–2675.

[37] J. Lagorse, M. G. Simoes, and A. Miraoui, "A multi-agent fuzzy logic based energy management of hybrid systems," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Edmonton, AB, Canada, 2008, pp. 1–7.

[38] J. Zeng, J. F. Liu, H. W. Ngan, and J. Wu, "A multi-agent solution to energy management of distributed hybrid renewable energy generated system," in *Proc. 8th Int. Conf. Advances Power Syst. Control, Oper. Manage.*, Hong Kong, China, 2009, pp. 1–6.

[39] R. Roche, L. Idoumghar, B. Blunier, and A. Miraoui, "Optimized fuel cell array energy management using multi-agent systems," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Orlando, FL, 2011, pp. 1–8.

[40] R. Belkacemi, A. Feliachi, M. A. Choudhry, and J. E. Saymansky, "Multi-agent systems hardware development and deployment for smart grid control applications," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, San Diego, CA, 2011, pp. 1–8.

[41] G. G. Karady, A. A. Daoud, and M. A. Mohamed, "On-line transient stability enhancement using multi-agent technique," in *Proc. IEEE Power Eng. Soc. Winter Meeting*, 2002, vol. 2, pp. 893–899.

[42] G. G. Karady and M. A. Mohamed, "Improving transient stability using fast valving based on tracking rotor-angle and active power," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, Chicago, IL, USA, 2002, vol. 3, pp. 1576–1581.

[43] C. Dou, C. Mao, Z. Bo, and X. Zhang, "A multi-agent model based decentralized coordinated control for large power system transient stability improvement," in *Proc. 45th Int. Univ. Power Eng. Conf.*, Cardiff, U.K., 2010, pp. 1–5.

[44] I. Praca, C. Ramos, Z. Vale, and M. Cordeiro, "MASCEM: A multiagent system that simulates competitive electricity markets," *IEEE Intel1. Syst.*, vol. 18, no. 6, pp. 54–60, Nov./Dec. 2003.

[45] S. E. Widergren, J. M. Roop, R. T. Guttromson, and Z. Huang, "Simulating the dynamic coupling of market and physical system operations," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2004, pp. 748–753.

[46] T. Li, Z. Xiao, M. Huang, J. Yu, and J. Hu, "Control system simulation of micro grid based in IP and multi-agent," in *Proc. Int. Conf. Inf., Netw. Autom.*, Kunming, China, 2010, pp. 235–239.

[47] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Proc. IEEE Power Eng. Soc. Power Syst. Conf. Expo.*, Seattle, WA, 2009, pp. 1–8.

[48] H. N. Aung, A. M. Khambadkone, D. Srinivasan, and T. Logenthiran, "Agent-based intelligent control for real-time operation of a micro grid," in *Proc. Joint Int. Conf. Power Electron., Drives Energy Syst. and Power India*, New Delhi, 2010, pp. 1–6.

[49] T. Logenthiran, D. Srinivasan, and D. Wong, "Multi-agent coordination for DER in micro grid," in *Proc. IEEE ICSET Int. Conf. Sustain. Energy Technol.*, Singapore, 2008, pp. 77–82.

[50] T. Logenthiran, D. Srinivasam, A. M. Khambadkone, and H. N. Aung, "Scalable multi-agent system (MAS) for operation of a micro grid in islanded mode," in *Proc. Joint Int. Conf. Power Electron., Drives Energy Syst. and Power India*, New Delhi, India, 2010, pp. 1–6.

[51] J. Lagorse, D. Paire, and A. Miraoui, "A multi-agent system for energy management of distributed power sources," *Renew. Energy*, vol. 35, no. 1, pp. 174–182, Jan. 2010.

[52] S. Rahman, M. Pipattanasomporn, and Y. Teklu, "Intelligent distributed autonomous power systems (IDAPS)," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Tampa, FL, 2007, pp. 1–8.

[53] J. Rocabert, G. M. S. Azevedo, A. Luna, J. M. Guerrero, J. I. Candela, and P. Rodriguez, "Intelligent connection agent for three-phase grid-connected micro grids," *IEEE Trans. Power Electron.*, vol. 26, no. 10, pp. 2993–3005, Oct. 2011.

[54] M. K. Kouluri and R. K. Pandey, "Intelligent agent based micro grid control," in *Proc. Int. Conf. Intell. Agent Multi-agent Syst.*, 2011, pp. 62–66.

[55] O. A. Mohammed, M. A. Nayeem, and A. K. Kaviani, "A laboratory based micro grid and distributed generation infrastructure for studying connectivity issues to operational power systems," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Minneapolis, MN, 2010, pp. 1–6.

[56] J. Oyarzabal, J. Jimena, J. Ruela, A. Engler, and C. Hardt, "Agent based micro grid management system," in *Proc. Int. Conf. Future Power Syst.*, Amsterdam, the Netherlands, 2005, pp. 1–6.

[57] T. Logenthiran, D. Srinivasan, A. M. Khambadkone, and H. N. Aung, "Multiagent system for real-time operation of a micro grid in real-time digital simulator," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 925–933, Jun. 2012.

[58] S. Strassburger, "On the HLA-based coupling of simulation tools," in *Proc. Eur. Simul. Multiconf.*, Warsaw, Poland, 1999, pp. 45–51.

[59] R. M. Fujimoto, "Time management in the high level architecture," *Simulation*, pp. 388–400, Dec. 1998.

[60] J. H. Jeon, J. Y. Kim, H. M. Kim, S. K. Kim, C. Cho, J. M. Kim, J. B. Ahn, and K. Y. Nam, "Development of hardware in-the-loop simulation system for testing operation and control functions of microgrid," *IEEE Trans. Power Electron.*, vol. 25, no. 12, pp. 2919–2929, Dec. 2010.

[61] G. Nguyen, T. T. Dang, L. Hluchy, M. Laclavik, Z. Balogh, and I. Budinska, "Agent platform evaluation and comparison," Bratislava, Slovakia, Tech. Rep. Pellucid 5FP IST-2001-34519, 2002, pp. 1–11.

[62] M. Malowidzki, "Network simulators: A developer's perspective," in *Proc. Int. Perform. Eval. Comput. Telecommun. Syst. SPECTS' 04*, San Jose, CA, USA, 2004, pp. 1–9.

[63] H. Lin, S. Sambamoorthy, S. Shukla, J. Thorp, and L. Mili, "Power system and communication network co-simulation for smart grid applications," in *Proc. IEEE Power Eng. Soc. Conf. Innovat. Smart Grid Technol.*, Anaheim, CA, USA, 2011, pp. 1–6.

[64] D. Pawlaszczyk, "Scalable multi-agent based simulation-considering efficient simulation of transport logistic networks," in *Proc. 12th ASIM Conf.–Simul. Prod. Logist.*, Sep. 26–27, 2006, pp. 479–488.

[65] J. Huang, M. Tung, K. M. Wang, and M. Lee, "Smart time management-the unified time synchronization interface for the distributed simulation," *Comput. Std. Interfaces*, vol. 27, pp. 149–161, 2005.

[66] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules*, IEEE Standard 1516, 2010.

[67] A. Helleboogh, T. Holvoet, D. Waynes, and Y. Berbes, "Extending time management support for multi-agent systems," in *Proc. Multi-Agent and Multi-Agent-Based Simulation: Joint Workshop*, New York, 2004, pp. 37–48.

[68] *"RTI 1.3-Next Generation Programmer's Guide,"* ver. 5, Department of Defense, Defense Modeling and Simulation Office, 2002.

[69] Microsoft, Interprocess Communication. 2012. [Online]. Available: http://msdn.microsoft.com