9th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 22-24, 2008

Agent-Based Systems in Power System Control

Cristinel COSTEA[1], Adrian PETROVAN[2]
North University of Baia Mare
str.V.Babes nr.62A, RO-430083 Baia Mare
[1]ccostea@ubm.ro, [2]adrian.petrovan@ubm.ro

Abstract—Cooperation become more and more popular in software industry and the emergent agent technology was also applied to industrial systems. The paper attempts to summarize some convergent developments in power system area that exploit agent technology and present a simple simulation based on Matlab and Java Agent DEvelopment Kit.

***Index Terms*—Power Systems Control, Multi-Agent Systems**

I. INTRODUCTION

Power networks are operated by thousands of devices following simple rules with local information. Some of these control devices are already pre-programmed for anticipated situations, but the liberalization of electricity markets or new trends (such as distributed generation, renewable energy) increase interconnectivity between the components and the centralized real-time control becomes more difficult.

As networking and embedding technologies advance, it is more possible to design autonomous entities for such process automation functions where the tasks require cooperative distributed problem solving.

Agents can improve the control devices, such as relays, voltage regulators or Flexible AC Transmission Systems (FACTS) devices. The connection between software entities and automation subsystems are fixed (generally defined at design-time), but the systems should deal with unanticipated requests.

The models for the architecture of future electricity systems consider new concepts: primary the entire system (transmission and distribution) can be designed and used as an integrated unit. The network must provide connectivity (multiple links) between points of power supply and demand. The network must interact permanently with the customers who can optimize his options and should pay for required services.

II. MULTI-AGENT SYSTEM TECHNOLOGY

An agent is an abstraction object (software or hardware) capable of autonomous action in some environment; in other words “is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors,” [25].

The agent concept is similar with software object, but Object-Oriented Programming (OOP) defines the object in terms of methods and attributes, an agent is defined in terms of behavior and ontology. Behaviors implement the tasks (or intentions) of an agent. They are logical activity units that can be composed in various ways to achieve complex execution patterns and that can be concurrently executed. The ontology indicates the vocabulary of the symbols used in the messages content. The agents involves in a communication must ascribe the same meaning to these symbols for the communication to be effective.

A stronger notion of agent include mental properties [31], like knowledge, belief or intention; additionally maybe consider mobility- agents can move from one host to another, veracity- agents do not knowingly communicate false information, benevolence- agents always try to do what they are asked of, rationality- agents will try to achieve their goals.

In Multi-Agent System (MAS) tasks are deployed by interacting agents that can cooperate with each other. Some characteristics generally adopted: “each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint; there is no system global control; data are decentralized; and computation is asynchronous” [27]. MAS can be considered "self-organized systems" as they tend to find the best solution for their problems without external intervention.

Some challenges in developing MAS are task decomposition, defining the agent behavioral rules, agent coordination, setting the environment where agents live.

III. MAS IN POWER SYSTEM

The industrial processes are susceptible to power quality problems outside the scope of conventional interruptions. The information is critical in detecting, resolving and preventing power quality problems and is essential for evaluating total system performance. A power-quality agent can read the measured data from the process, compute power quality indexes, and shared the processed data with other agents. A distributed system is more available then one centralized to detect incidents like transients (frequency and amplitude changes in voltage and/or current waveforms), sags and swells, flicker, noise, harmonics or frequency variations.

The actually electric networks are operated local and distributed by heterogeneous agents that can range from simple devices (like relays) to intelligent entities (like human operators). We can say that each agent can sense only a few of the network’s state variables and influence only a few of its control variables.

Several researches have been made using agents in power system applications, the typical areas referring to restoration and expansion planning, cascading failures, optimal power flow, secondary voltage control, oscillation damping control, adaptive relaying, fault diagnosis [32], power market or Strategic Power Infrastructure Defense [16].

Similar to classic hierarchical operations, in [1] [11] the agents form layers: the reactive layer consists of agents that perform pre-programmed self-healing actions that required an immediate response such as protection agents, fault isolations, frequency stability agents or generation agents. The agents in coordination layer (middle) identify which triggering event from the reactive layer is urgent, important or resource consuming. They update system's model and check if the commands from top layer represent the current status. If the plans do not match middle layer trigger the deliberative layer to modify the plans. The deliberative layer consists of cognitive agents including restoration agents, planning agents or failure monitoring agents [1].

The reactive agents are not necessary intelligent, but its main ability is to react fast considering each and every situation in advance. The cognitive agents have a knowledge base, handle interactions with other agents and have goals and plans.

The current practice in outages or faults in distribution power systems are isolation of the faulted part of the system, thus some unfaulted areas lose power. An efficient switch operation scheme who restore the power to an optimal target network configuration is referred to power system restoration.

The problem can be formulated as an objective function satisfying the system constraints, where the objective is to maximize the supply of power to as many loads as possible. A constraint can limit the generation, cable limits (flow limits from one node to another), radial configuration (only one branches feed the node). There are many algorithms and most of the restoration techniques are centralized, but the decentralized approaches also exist.

Petcu [24] consider that restoration problem can be formulated as a constraint optimization problem. Constraint Satisfaction Problem (CSP) is a problem to find a configuration (consistent assignment of values to variables) that satisfies the given conditions (constraints). A typical example is the well known n-queens problem. Additionally, in Distributed CSP (DCSP) variables and constraints are distributed among multiple automated agents, many problems being formalized as DCSP and solved with distributed algorithms (asynchronous backtracking, asynchronous weak-commitment search, distributed dynamic backtracking [34]).

Nagata et al. [21], [22] obtained interesting results with multi-agents for restoration problem. The authors started with two types of agents, Load Agents (LAG) and Feeder Agents (FAG), which can act at higher levels. After the switches are open in the de-energized area, each LAG sends a request for restoration, and FAG stores all the requests and forwards them to other FAGs, which thereby initiate the closing of load switches by LAGs in order to restore. This technique is based on agent negotiation, but is difficult to compare such solution with the global optimum.

A cascading failure is a set of switching operations that automatic de-energize equipments, leaving the grid with less ability to generate or deliver electric energy [9]. The switching operations are triggered by those changes in the continuous variables that produce excessive stresses in some generation and delivery devices. Relays sense these excesses, and react to de-energize the threatened devices.

Talukdar and Camponagara presented in [5] [28] an algorithm based on decomposition for mapping network-control-problems into collaborative nets using agents distributed through the network: formulate the network control problem as a series of static optimization problems, decompose (*P*) into a set of sub-problems, assign each sub-problem to an agent, provide a protocol for the agents to use in collaborating with their neighbors.

In the decomposition of problem (*P*) into *m* sub-problems was considered that most of the equality or inequality constraints from electric grid are sparse (expressed in terms of only few variables) and local (variables come from a small geographic area), each sub-problem (assigned to one agent) contains a part of the objective of (*P*) and some of its constraints. The agents will work on their sub-problems asynchronously.

The objective of the reactive power dispatch [17][35] is to minimize the active power loss in the transmission network which can be described as an optimization problem subject to voltage constraints(voltage magnitude at each node must lie within permissible ranges) and power flow constraint(power flow of each branch must lie within their permissible ranges). For solving the problem a multi-agent based particle swarm optimization approach (MAPSO), was adopted, and considered an agent a particle from the Particle Swarm Optimization and a candidate solution to the optimization problem. PSO is a metaheuristic optimization method introduced by Eberhart (1995), based on a number of particles "flying" around in a multidimensional search space looking for the global extreme. Each particle modifies its position according to its own experience, and the experience of neighboring particles, making use of the best position encountered by itself and its neighbors (Fig.1). The swarm direction of a particle is defined by the set of particles neighboring to the particle and its history experience.

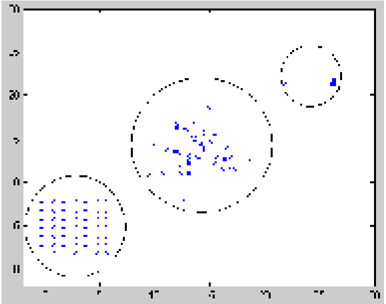


Figure 1. Matlab simulation of a swarm in 2D searching for a global minimum (from W. Elshamy, www.mathworks.com).

Each agent firstly competes and cooperates with its neighbors to diffuse its useful information to the whole environment, and it can also use evolution mechanism of PSO to improve its knowledge; three operators are design for evolution: competition and cooperation operator, PSO operator and self-learning operator [35].

Flexible Alternating Current Transmission Systems (FACTS) devices affect power flow and/or node voltages dynamically, increase the controllability in power systems and offer opportunities to agent approaches. Typical fields of FACTS application are dynamic reactive power compensation, dynamic power flow control and power flow dispatching, increase of transmission capacity, improve of steady-state stability (voltage regulation) and enhancing dynamic stability (oscillation damping).

IV. DISTRIBUTED GENERATION AND MICROGRIDS

Distributed generation consists of different sources of electric power connected to the distribution network or to a customer site, a concept approach distinct from the traditional central-plant model for electricity generation and delivery. Examples of DG are photovoltaic solar systems, small and medium-scale wind turbine farms or combined generation of heat and power (CHP). Increasing interest for these devices will change the distribution networks who expect to evolve from a hierarchically controlled structure into a network of networks (similar to Internet concept), in which a vast number of system parts communicate with and influence each other.

In market-based control [29], many control agents competitively negotiate and trade on an electronic market to optimally achieve their local control action goals. Each device are supervised by a control agent who tries to operate the device process in an economically optimal way conform to process's constraints. The agents negotiate prices on an electronic exchange market either their consumption or electricity production, and the resulting price determines the power volume allocated to each device.

The MAS protection coordination system presented in [30] consists of relay agents, distributed generator agents and equipment agents. The agents can communicate with each other within the same agent society or within different agent societies. By example, a distributed generator agent communicates with relay agent to provide connection status.

Distributed sources control conduct to problem of placing the generation near the loads and the microgrid concept.

MicroGrids are small electrical distribution systems that connect multiple customers to multiple distributed sources of generation and storage. A microgrid comprises a LV (up to 1 kV) or MV (1–69 kV) locally-controlled cluster of Distributed Energy Resources that behaves, from the grid's perspective, as a single producer or load unit. A microgrid operates safely and efficiently within its local distribution network, but it is also capable of islanding. [8]

One of the critical components is the switch [15]. The static switch can autonomously island the feeders when a disturbance occurs; after the event is no longer present the reconnection of feeder can be realized also autonomously.

Some concepts used in microgrids design and related to agent-based approach are peer-to-peer and plug-and-play.

The peer-to-peer concept insures that none of component is the master controller or central storage; there are no protection components such as a master coordinator or communication system; thus the microgrid can continue operate with loss of any component.

Plug-and-play implies that a unit can be placed in any node of the electrical grid without reengineering the whole system.

V. INTELLIGENT ELECTRONIC DEVICES

Intelligent Electronic Devices (IEDs) are the standard [2] in new or upgraded integrated substation protection. Microprocessor based protective relays protect substation equipment, but also provide additionally functions like metering, event recording or built in fault analysis tool. They monitor the state of their environment and can adjust their operations to changes; being viewed as reactive systems, they are expected to identify changes in the environment and to react to a predefined set of events [18].

Such devices of different types interact between themselves and with other levels of the substation automation system in distributed applications.

The new IEC 61850 international standard for substation communications allow [2] future development and implementation of self-configuring substation automation systems. IEC 61850 defines functions of a substation automation system (SAS) related to the protection, control, monitoring and recording of the equipment in the substation; these functions are implemented within a single IED or can be distributed between multiple devices using communications interface, thus different multifunctional devices can be considered multi-agent systems themselves.

Embedded software agents are an artifact of the need to exhibit some degree of autonomy and mobility (should support the ability to move code or data from one platform to another). The hardware environment (IED) must provide necessary computational and I/O capabilities needed to support a software agent [30]. The agent takes sensory inputs, such as local measurements of the current, voltage, and breaker status and produces as output actions such as breaker trip signals, adjusting transformer tap settings or switching signals in capacitor banks.

In last period an increasing interest was manifested in virtual machines such as .NET Micro Framework or Java Virtual Machine. An implementation of JVM in low-cost FPGA is Java Optimized Processor (JOP). According to [29] JOP is the smallest hardware realization of the JVM available to date, and the results using compiled Java for a RISC processor in FPGA, with a similar resource usage and maximum clock frequency to JOP, showed that native execution of Java byte-codes is faster than compiled Java.

Java processors like JOP will increase the acceptance of Java agents in small embedded real-time systems.

MAS PLATFORMS

The electric networks are operated local and distributed by heterogeneous agents that can range from simple devices (like relays) to intelligent entities (like human operators). We can say that each agent can sense only a few of the network's state variables and influence only a few of its control variables.

The MAS literature [4] indicates a large number of agent-oriented frameworks and the most important standardization effort in the software agent is Foundation for Intelligent Physical Agents [6].

FIPA specifications establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents and suggests mandatory components or normative agents such as directory facilitator, agent management system or agent communication channel.

There are a variety of the FIPA implementations (ZEUS, RETSINA, FIPA-OS, GRASSHOPER). One of them is JADE - an open-source middleware (Java Agent DEvelopment Framework [12]) used over the last years by academic and industrial organizations. It support interoperability by adhering to the FIPA [6] specifications, and can be used on devices with limited resources [4]. The extension JADE-LEAP (Lightweight Extensible Agent Platform) is capable of running under Java2ME.

JADE agents are executed in a run-time environment called an agent container. Each agent runs on its own thread and agents within a container share resources and services (e.g., thread scheduling and messaging support). Typically, in a multi-agent application, there will be several containers (with agents) running on different machines with different operating systems. The first container started must be a main container which maintains a central registry of all the others so that agents can discover and interact with each other. The configuration can be controlled by a remote Graphical User Interface (Fig.2).

A distributed JADE platform is composed of several run-time containers launched on one or more hosts on a network. Each hardware device typically executes one Java Virtual Machine (JVM), which can host one or more JADE instances; inter-JVM communication is based on JAVA Remote Method Invocation (RMI).

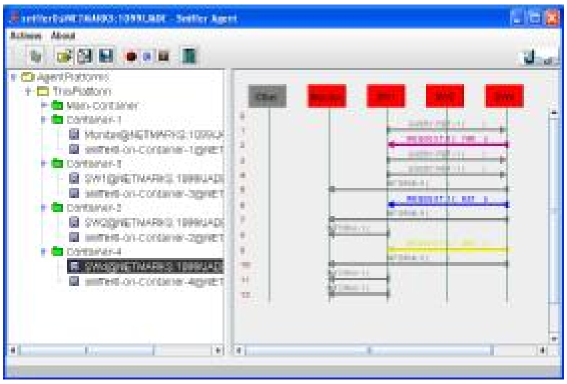


Figure 2. JADE Graphical User Interface (sniffer agent).

Agent communication is performed through message transfer; message representation is based on the Agent Communication Language (ACL) formulated by FIPA [6], a language based on speech act theory containing two distinct parts: the communicative act and the content of the message. JADE creates and manages a queue of incoming ACL messages, private to each agent; agents can access their queue via several modes: blocking, polling, timeout and pattern matching based.

The run-time includes also some ready to use behaviors for the most common tasks in agent programming such as interaction protocols, waking under certain condition and structuring complex tasks. There are a number of classes that extend the basic object Behavior, like SimpleBehavior, OneShotBehavior or CyclicBehavior. More functionalities and detailed description can be found in programmers's guide [12].

VI. CASE STUDY

Centralized control (supported by SCADA) may not be practical in large systems; an alternative strategy is the distribution of intelligence (more agents) and local interventions where the disturbance originates [1]. An agent may supervise a subsystem or device, but in the meantime it changes messages with other agents, creating a virtual agents-layer.

Multi-agent models are oriented toward autonomy and collaborations and planning a MAS simulation depend on both agent environment and the coordinated process simulation.

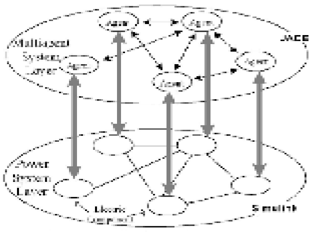


Figure 3. Layers concept.

In such cases the simulations are faced to a basic layer for process simulation (Matlab) and another layer for controllers (JADE) represented in Fig.3.

The first step of our simulations was to study the interfacing possibilities between the Java based agent framework and Simulink. Furthermore, the agents attached to breakers face to a simple problem: tries to detect and then isolate a current fault [26]. The agents only communicate with their neighbor agents: if a breaker agent senses the fault current, it will communicate with its downstream neighbors; if the downstream switch agent also senses the fault current, then the fault is not in the zone between their corresponding devices, so there is no decision to open the breakers (Fig. 4). If an agent senses the fault but its neighbor does not sense the fault, the agents consider a fault between its devices and both agents will lock out the breakers.

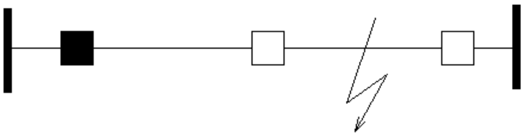
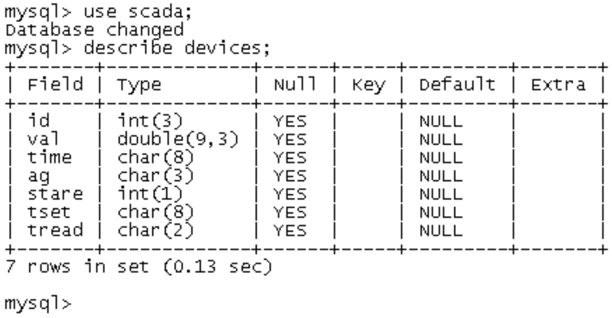


Figure 4. Fault detecting.

We considered a simple scenario from the local distribution system situated in Baia Mare (Fig. 6) and the Simulink model (Fig. 7, 8). In this simple scenario we dispose 2 sources (the second is reserve source), one load, 4 breakers and a fault. Each breaker can be controlled by its Java agent. Four values of measurement current can be read it by external Java programs through Matlab S-Functions (Sf01 block in Fig. 8) and MatlabServer [36].

The communications between Matlab and Java was realized in this simulation by a database record (MySQL in actual simulations). In each computing step, a breaker block from Simulink access the table 'devices' (table structure shown in Fig. 5) and read a state value from corresponding record and then write the current value of its device. Similar, the external JAVA agent senses the current values reading the same record (ID field identify a device) and setting the device state: 0 for open the breaker and 1 for close.



```
mysql> use scada;
Database changed
mysql> describe devices;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int(3)      | YES  |     | NULL    |       |
| val   | double(9,3) | YES  |     | NULL    |       |
| time  | char(8)     | YES  |     | NULL    |       |
| ag    | char(3)     | YES  |     | NULL    |       |
| stare | int(1)      | YES  |     | NULL    |       |
| tset  | char(8)     | YES  |     | NULL    |       |
| tread | char(2)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
7 rows in set (0.13 sec)

mysql>
```

Figure 5. MySQL table structure

External JADE agents can sense the current and control the breakers SW1 and SW4 (in this scenario SW3 is blocked). After a configured time, when breakers are opened, the agent will close SW2 to feed the load from second source.

A Sniffer Agent (component of JADE platform) allows for the tracking of messages exchanged between Monitor, SW1, SW2 and SW4 agents (Fig. 2).

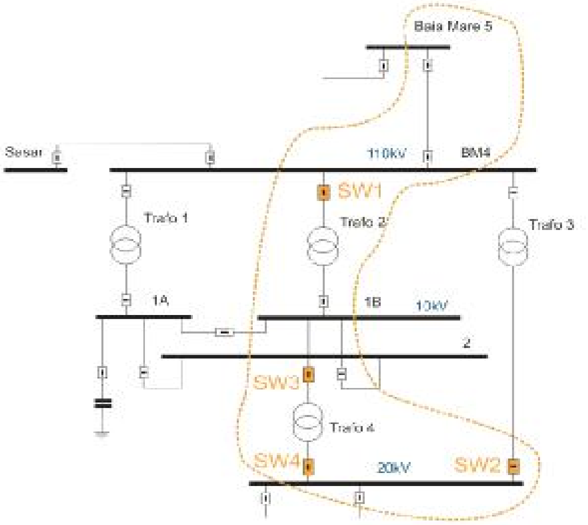


Figure 6. Real scenario

Fig. 9 shows the Matlab results of relays operations. Agents coordinate their actions to detect and isolate the fault, but maintaining power supply.

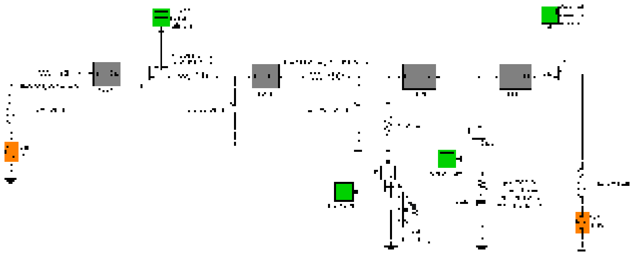


Figure 7. Simulink model

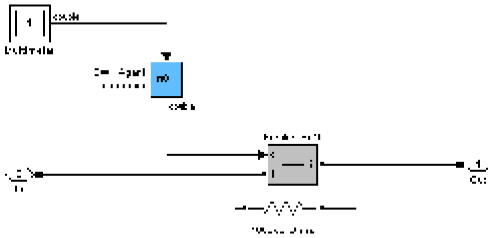


Figure 8. Breaker subsystems

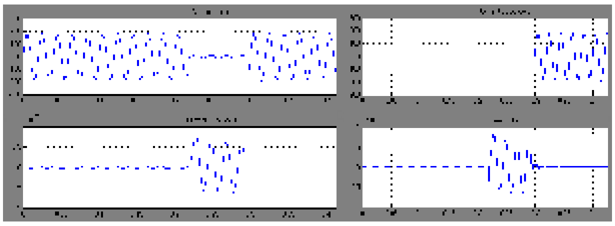


Figure 9. Agents coordinate their actions to detect and isolate the fault: load current (upper left), fault current (bottom right)

VII. CONCLUSIONS AND FUTURE WORK

Modern infrastructures contain many entities highly interconnected expected to involve an increasing level of intelligence and integration of information technology. The components of the power grid would be agents acting independently and cooperate to achieve global performance. This technology offer desirable properties over centralized systems like scalability, flexibility, reduced costs, separately development, re-usability.

Multi-agent models are oriented toward autonomy and collaborations, such simulations depending on both agent's environment and the coordinated process simulation.

This approach conduct to a new application under development in Java (Eclipse) and JADE for multi-agent simulations dedicated to power system controls. The present simulations permit us to refine the Java-Matlab communication routines and continue more realistic scenarios to develop multi-agent based algorithms.

In same time deficiencies exists. One of the most is the accidental increasing amount of communication between agents. Another issue of simulation is the first started container which maintains a central registry of agents. This is a single point of failure that is not in idea of decentralization, but can be surpassed in this stage.

REFERENCES

[1] M. Amin, "Toward Self-Healing Infrastructure Systems," Computer, vol. 33, no. 8, pp. 44-53, Aug., 2000

[2] A.Apostolov, "Multi-agent systems and IEC 61850" , IEEE Power Engineering Society General Meeting, 2006

[3] F. Bellifemine, G. Caire, D. Greenwood "Developing Multi-Agent Systems with JADE," 2007 John Wiley & Sons, Ltd

[4] Rafael Bordini, L. Braubach, M. Dastani, A. E. Fallah Seghrouchni, J. Gomez-Sanz, J. Leite, G. O'Hare, A. Pokahr, A. Ricci, "A Survey of Programming Languages and Platforms for Multi-Agent Systems", in: Informatica 30,2006

[5] Camponogara, E., "Controlling Networks with Collaborative Nets", Ph.D. Dissertation, ECE Department, Aug. 15th, 2000, [online] http://www.cs.cmu.edu/~camponog/

[6] FIPA, The Foundation for Intelligent Physical Agents http://www.fipa.org

[7] N. Hatziargyriou, H.Asano, R.Iravani, C. Marnay, Microgrids-An Overview of Ongoing Research, Development, and Demonstration Projects, IEEE Power & Energy Magazine, july/august 2007

[8] P. Hines, S. Talukdar, " Reducing the Costs of Disturbances to the Electric Power Network." Presented at the Int. Assn. for Energy Economics meeting, Washington D.C., June 2004.

[9] P. Hines, S. Talukdar, "Controlling Cascading Failures with Cooperative Autonomous Agents", unpublished.

[10] Qi Huang, K. Qin, W. Wang, "A software architecture based on multi-agent and grid computing for electric power system applications," Proceedings of the International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06)

[11] M. Ilic ,F.F. Wu, Research and Applications on Real-time Control of Power Grids: Past Successes and Future Opportunities, Bulk Power System Dynamics and Control - VI, August 22-27, 2004, Cortina d'Ampezzo, Italy

[12] JADE - Java Agent Development Framework, [online] http://jade.tilab.com.

[13] K. Kok, C. Warmer, and René Kamphuis, "The PowerMatcher: Multiagent Control of Electricity Demand and Supply", IEEE Intelligent Systems, March/April 2006

[14] B. H. Kim, Ross Baldick, "A Comparison of Distributed Optimal Power Flow Algorithms", IEEE Transactions on Power Systems, Vol.15, No.2, May 2000

[15] R. H. Lasseter, "Microgrids and Distributed Generation", Journal of Energy Engineering, American Society of Civil Engineers, Sept.2007

[16] Hao Li, G.W.RosenWald, J.Jung, Chen-Ching Liu, "Strategic Power Infrastructure Defense", Proceedings of the IEEE Volume 93, Issue 5, May 2005 Page(s): 918 – 933

[17] Zhongxu Li, and Yutian Liu, “Reactive Power Optimization Using Agent-based Grid Computing”

[18] Dan C. Marinescu, Yongchang Ji, Gabriela M. Marinescu, Xin Bai, “Physical Awareness and Embedded Software Agents” , Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices, Bologna, Italy, July 2002

[19] [online] http://www.mathworks.com

[20] [online] http://www.mysql.com

[21] T. Nagata, H. Sasaki, “A Multi-Agent Approach to Power System Restoration”, IEEE Transactions on Power Systems, Vol. 17, No. 2, May 2002

[22] T. Nagata, Y. Tao, H. Sasaki, H. Fujita, “Decentralized Approach to Power System Restoration by means of Multi-agent Approach”, Bulk Power System Dynamics and Control - VI, August 22-27, 2004, Cortina d’Ampezzo, Italy

[23] M. Pechoucek, S. Thompson: Agents in Industry:The Best from the AAMAS 2005 Industry Track. IEEE Intelligent Systems. 2006, p. 1541-1672.

[24] A.Petcu, B. Faltings.”Distributed generator maintenance scheduling” Proceedings of the First International ICSC Symposium on Artificial Intelligence in Energy Systems and Power: AIESP’06, Madeira, Portugal, Feb 2006.

[25] S.J.Russell, P. Norvig, “Artificial Intelligence: a Modern Approach”, Prentice Hall, 1995

[26] J.M.Solanki, N.N.Schulz, W.Gao, “Reconfiguration for Restoration of Power Systems Using a Multi-Agent System,” Proceedings of the 37th Annual North American Power Symposium, Iowa State University, Ames, Iowa, October 23-25, 2005

[27] K. Sycara,, Multiagent Systems , AI Magazine , 19(2): Summer 1998, 1998, 79-92 [online] http://www.aaai.org/AITopics/html/multi.html

[28] S. Talukdar, Camponagara, E., “Collaborative Nets”, HICSS-2000, Proceedings, Jan 2000, [online] http://citeseer.ist.psu.edu/471904.html

[29] M. Schoeberl, “Evaluation of a Java Processor,” In Tagungsband Austrochip 2005, Austrochip 2005, Vienna, Austria, October 2005, [online] http://www.jopdesign.com

[30] H. Wan, Li, K.K.; Wong, K.P.,"Multi-agent application of substation protection coordination with distributed generators",Future Power Systems, 2005 International Conference on, 16-18 Nov. 2005, Pages: 6 pp.

[31] M. Wooldridge, “An Introduction to MultiAgent Systems”, John Wiley & Sons, Ltd., 2001,2002

[32] J. Yang, M. Montakhab, A. Pipe, B. Carse and T.S. Davies, “Application of Multi-Agent Technology to Fault Diagnosis of Power Distribution Systems,” Proceedings of the 4th International ICSC Symposium on Engineering of Intelligent Systems, March 2004

[33] F. Ygge,H. Akkermans, "Making a Case for Multi-Agent Systems", Proceedings of the Eighth Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '97

[34] M. Yokoo, “Algorithms for Distributed Constraint Satisfaction: A Review”, Autonomous Agents and Multi-Agents Systems, Vol.3, No.2, 2000, pp.198-212.

[35] B. Zhao,C.X.Guo,Y.J.Cao, “A Mutiagent-based Particle Swarm Optimization Approach for Optimal Reactive Power Dispatch”, IEEE Transactions on Power Systems, Vol.20, No.2, May 2005

[36] [online]http://plasmapowered.com/ ,Calling_MatLab_from_Java