



Graduate Theses, Dissertations, and Problem Reports

2008

Application of multi-agents to power distribution systems

Koushaly Nareshkumar
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Nareshkumar, Koushaly, "Application of multi-agents to power distribution systems" (2008). *Graduate Theses, Dissertations, and Problem Reports*. 4409.
<https://researchrepository.wvu.edu/etd/4409>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Application of Multi-Agents to Power Distribution Systems

by

Koushaly Nareshkumar

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Prof. Ali Feliachi, Ph.D.
Prof. Hong-Jian Lai, Ph.D.
Prof. Muhammad A. Choudhry, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2008

Keywords: Multi-Agent System, Fault Detection, Power System Reconfiguration, DEW© , JADE

Copyright 2008 Koushaly Nareshkumar

ABSTRACT

Application of Multi-Agents to Power Distribution Systems

by

Koushaly Nareshkumar

Master of Science in Electrical Engineering

West Virginia University

Professor Muhammad Choudhry, Ph.D., Chair

The electric power system has become a very complicated network at present because of re-structuring and the penetration of distributed energy resources. In addition, due to increasing demand for power, issues such as transmission congestion have made the power system stressed. A single fault can lead to massive cascading effects, affecting the power supply and power quality. An overall solution for these issues can be obtained by a new artificial intelligent mechanism called the multi-agent system. A multi-agent system is a collection of agents, which senses the environmental changes and acts diligently on the environment in order to achieve its objectives. Due to the increasing speed and decreasing cost in communication and computation of complex matrices, multi-agent system promise to be a viable solution for today's intrinsic network problems.

A multi-agent system model for fault detection and reconfiguration is presented in this thesis. These models are developed based on graph theory and mathematical programming. A mathematical model is developed to specify the objective function and the constraints.

The multi-agent models are simulated in Java Agent Development Framework and Matlab® and are applied to the power system model designed in the commercial software, Distributed Engineering Workstation©. The circuit that is used to model the power distribution system is the Circuit of the Future, developed by Southern California Edison.

The multi-agent system model can precisely detect the fault location and according to the type of fault, it reconfigures the system to supply as much load as possible by satisfying the power balance and line capacity constraints. The model is also capable of handling the assignment of load priorities.

All possible fault cases were tested and a few critical test scenarios are presented in this thesis. The results obtained were promising and were as expected.

DEDICATION

I would like to dedicate this thesis to
My lovely mother, who raised me to be who I am now,

&

My loving husband and greatest friend Naresh

Father, you are deeply missed

ACKNOWLEDGEMENT

I take this opportunity to sincerely thank my advisor, Professor Muhammad Choudhry, who guided, supported and advised me from the very first day at WVU. His guidance and training helped me a lot in gaining confidence and knowledge right throughout my studies at WVU from the time when I was instructing the power laboratory and throughout my research work. I also would like to express my heartfelt thanks to Professor Ali Feliachi, who had faith in me and supported and advised me throughout my studies and research work. My deep gratitude also goes to my mathematics advisor, Professor Hong-Jian Lai, who gave me insight of the mathematical models required for my work.

I would like to thank Professor Hui Wan, all the other Faculty members in APERC and the staff in LDCSEE for their kind support and encouragement. My thanks also go to my friends in APERC for the healthy discussions and for making my time enjoyable. A Special thanks to Pinak Tulpule in helping me understand the shipboard Energy Management System and to Summiya Moheuddin for helping me with JADE. My sincere thanks go to all my friends in Morgantown for their kindness and the encouraging words.

I would like to thank my mother and sister, who had given me constant moral support and encouragement from Sri Lanka. I also thank my parents-in-law for their encouragement and advice. Last but not least I express my thanks to my husband for his tremendous support, patience and understanding throughout my studies, without which I would not have achieved this goal.

This work was supported in part by grants from the US DEPSCoR/ONR grant No. N00014-03-1-0660 and the US DoE grant No. DE-FC26-06NT42793.

CONTENTS

ABSTRACT	ii
Dedication.....	iii
Acknowledgement	iv
Contents	v
List of Tables	viii
List of Figures	ix
Nomenclature	xi
Chapter 1	1
Introduction.....	1
1.1 Background	1
1.2 Introduction to Multi-Agent System.....	4
1.3 Agent Communication	6
1.3.1 FIPA - ACL	7
1.3.2 Knowledge Query and Manipulation Language	8
1.4 Problem Statement.....	9
1.5 Approach	9
1.6 Thesis Outline.....	10
Chapter 2	11
Literature Review.....	11
2.1 General Overview of MAS in Power System	11
2.2 MAS in Automatic Fault Detection.....	13

2.3	Reconfiguration & Restoration by MAS	14
2.4	MAS in Distribution System with DER	17
2.5	Protection Coordination	18
2.6	Voltage Control.....	20
2.7	Technical Challenges	21
2.8	Other MAS Applications	22
2.8.1	Highway Traffic Management.....	23
2.8.2	Agent Based Air Traffic Control.....	23
2.8.3	Mine Detection	24
2.9	Conclusion.....	24
Chapter 3	25
Software Packages.....		25
3.1	Matlab®.....	26
3.2	Distributed Engineering Workstation© (DEW).....	27
3.2.1	Description of DEW	27
3.2.2	DEW Modeling Features	29
3.2.3	Pros and Cons of modeling with DEW in this work	30
3.3	Java Agent Development Framework	31
3.3.1	JADE Architecture	32
3.3.2	JADE Programming.....	34
3.4	Conclusion.....	35
Chapter 4	36
The Mathematical Model for Multi-Agent System.....		36
4.1	Mathematical Model	36
4.1.1	Graph Theory	36
4.1.2	Graph Theory representation of Power System	37

4.2	Fault Detection Algorithm	39
4.3	Reconfiguration & Restoration Algorithm	39
4.4	Illustration	42
4.5	Conclusion.....	43
Chapter 5		44
Simulation and Results.....		44
5.1	Circuit of the Future	46
5.2	JADE Fault Detection Simulation	48
5.2.1	Test Case 1: For a single fault	50
5.2.2	Test Case 2: For Multiple Faults	51
5.3	Matlab Reconfiguration Simulation.....	52
5.3.1	Illustration of Reconfiguration with IntelliTEAM® II	52
5.3.2	Test Case 3: For a Line Fault.....	57
5.3.3	Test Case 4: Shortage of Source Capacity	57
5.4	Algorithm Execution Duration	60
5.5	Testing the Voltage Profile	61
5.6	Conclusion.....	62
Chapter 6		63
Conclusion and Future work.....		63
6.1	Conclusion.....	63
6.2	Future Work.....	65
References		67

LIST OF TABLES

Table 1-1: FIPA ACL Message Parameters [7].....	8
---	---

LIST OF FIGURES

Figure 1-1: A Multi-agent System in the sense of Power System.	5
Figure 2-1: SWOT matrix for current Agent Applications to power systems [9].	12
Figure 3-1: Architecture of the Simulation Model.	26
Figure 3-2: Snapshot of DEW Working Environment.	28
Figure 3-3: Major Functional Parts of DEW [57].	29
Figure 3-4: FIPA Agent Management Reference Model.	31
Figure 3-5: Snap shot of JADE run-time environment.	33
Figure 3-6: Snap shot of JADE GUI.	34
Figure 4-1: Diagram of the Notations of a Graph.	38
Figure 4-2: Graphical Representation of Simplified CoF.	43
Figure 5-1: Generic model of the MAS application in Power Distribution System.	45
Figure 5-2: Architecture of the Proposed MAS.	45
Figure 5-3: The Circuit of the Future modeled in DEW.	47
Figure 5-4: Flow chart for Fault Detection.	49
Figure 5-5: JADE Message Passing to identify the fault location for a single fault in N_1N_2	50
Figure 5-6: Sniffer Agent GUI for Multiple Fault Scenarios.	51
Figure 5-7: Output screen of the solution for Test Case 2.	51
Figure 5-8: IntelliTEAM® II under normal operations.	53
Figure 5-9: Reconfigured IntelliTEAM® II after the fault in the primary supply line.	53
Figure 5-10: Flow Chart for Reconfiguration.	54
Figure 5-11: GUI for user interaction for Reconfiguration.	55

Figure 5-12: Graphical representation of the CoF under normal operations.....	56
Figure 5-13: Reconfigured System after a line fault.....	58
Figure 5-14: Reconfigured Circuit during a shortage of source capacity.	59
Figure 5-15: Plot of Execution Time for Fault Detection Algorithm.	60
Figure 5-16: Plot of Execution Time for Reconfiguration Algorithm.	60
Figure 5-17: Voltage profile of the CoF under normal operations.	61
Figure 5-18: Voltage profile of the CoF under the faulty situation in Test Case 3.....	62

NOMENCLATURE

ACL	Agent Communication Language
CoF	Circuit of the Future
DER	Distributed Energy Resources
DEW	Distributed Engineering Workstation
DG	Distributed Generators
FIPA	Foundation for Intelligent Physical Agents
GUI	Graphical User Interface
JADE	Java Agent Development Framework
KQML	Knowledge Query and Manipulation Language
LAG	Load Agent
MAS	Multi-agent System
OOP	Object Oriented Programming
SAG	Switch Agent
SCE	Southern California Edison
WVU	West Virginia University

Chapter 1

INTRODUCTION

1.1 Background

The history of electricity goes back nearly two thousand years, starting from rubbing various substances, invention of light bulbs by Edison, a radical transformation of mechanical age to electrical age and so on. During the initial stages of commercial use of electric power, the power could not be transmitted over a long distant from a generating source, due to the fact that the generation was in direct current, which could not be easily stepped up over a long distance transmission. During this time, loads with different voltage levels were served by different generating sources.

In 1888, Tesla discovered the alternating current generation [1], which paved the way to the modern electric grid where multiple generating sources were interconnected with the same frequencies. This enormous invention led to the spread of electricity from the generating sources through long transmission lines, stepping down the voltages in substations then finally distributing to the end users through distribution lines. Electricity was produced using hydro plants, coal power plants, diesel generators, and then later, nuclear power plants etc. A few decades ago, some sources of renewable energy, such

as solar, wind, micro hydro, dendro (energy crop), cogeneration, micro turbines, etc were added to the generating sources. Some of these renewable sources have been installed at the load ends, in the distribution system, which are called the Distributed Energy Resources (DER).

The main reason for installing the DER as near as possible to the load centers is that the electric usage has become so vast that the present transmission capacity is not adequate to transfer the generation to the load centers. Also due to deregulation in the power industry, issues such as transmission congestion have caused a lot of reliability issues. By installing the DERs near the load centers, the grid does not require immediate expansion of long distance transmission lines. Reduction in the transmitted power, not only reduce the power loss in transmission and distribution, but also helps in delaying the new power transmission projects, which require long waiting time for approval and construction and thus postpones the immediate bulk investment.

However, interconnection of the DERs to the existing system makes the already meshed power grid, a more complicated network that needs a lot of considerations on issues like synchronization with the existing power grid, reliability etc. Distributed Generators¹ (DGs) can be connected to the system, when there is inadequacy in real or reactive power or they can be designed to supply the system in a continuous manner. In either case, it has to be ensured that after it is switched on into the system, the voltages and stability have to be maintained within the limits. It is also required that the protection coordination have to be modified greatly when incorporating the DGs into the system.

Electric power grid has become very complex and this, at times, can lead to cascading failures due to a single fault in transmission lines. The massive Northeast Blackout in 2003 is a very good example for this type of failure. This widespread power outage left around 50 million people in Canada and the United States without power and caused about 6 billion USD of financial losses [2]. The root cause of this major failure was due to high electric demand in an area that caused the high voltage transmission lines to go over their capacity limits and sagged and touched the overgrown trees in Ohio Service

¹ DGs and DERs are used interchangeably in this thesis

area. This caused cascading failures of the neighboring lines and ultimately, shutting down 100 generators. Similar failures, in smaller scales have happened in the recent past in other different areas.

Hence, it is vital to find ways to improve the reliability of the power system due to the increased power demand and congestion issues and to mitigate cascading power failures. By reducing power failures and maintaining the voltage profile to the end users, the power quality can be enhanced. This can be achieved by finding solutions that are more decentralized, automated, and more sophisticated with less human intervention. One of this is application of multi-agents to monitor and control power system in a decentralized manner to enhance the reliability of power supply. That is, during a line fault or shortage of power supply, to find solutions in a decentralized manner to serve as much load as possible to the consumers by re-routing the power flow or by islanding a portion of the load, which can be supplied by local DERs in order to continuously supply power to the end users without any or less interruptions.

Drastic reduction of cost and increasing speed in the emerging technologies in computing and communication have also paved the path to the usage of the most sophisticated control technologies, such as MAS. The researchers in the power engineering field are trying to exploit the advantages of these technologies and apply them to the power system in order to better serve the society.

Conventionally, many applications in power system were solved by human actions by obtaining information from the system. These same actions when operated by agents, can give timely and more reliable decisions with less human intervention. In addition, agents can learn from their previous experiences and act accordingly during different fault scenarios.

1.2 Introduction to Multi-Agent System

The objective of this section is to clearly understand what a MAS is. A few definitions of an agent are given below.

The Oxford English Dictionary presents an agent as,

“A person who provides a particular service, typically liaising between two other parties *or* a person or thing that takes an active role or produces a specified effect”

According to Webster English Dictionary, an agent is,

“An active and efficient cause; capable of producing a certain effect *or* A substance that exerts some force or effect *or* A representative who acts on behalf of other persons or organizations”

In computing terms, the dictionary of computing articulates an agent as,

“In the client-server model, the part of the system that performs information preparation and exchange on behalf of a client or server. Especially in the phrase ‘intelligent agent’ it implies some kind of automatic process which can communicate with other agents to perform some collective task on behalf of one or more humans” [3]

It has to be noted that there is no universally accepted definition for an agent. The definitions given to an agent, in terms of MAS, by two eminent authors are given below.

“An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives” – Wooldridge and Jennings (1995) [4]

“An agent is an autonomous computational entity such as a software program that can be viewed as perceiving its environment through sensors and acting upon this environment through its effectors” – Weiss [5]

Hence an Agent, in terms of this thesis, is a software entity that senses the changes in its environment and acts upon it to achieve its intended objectives. A Multi-agent System is a system composed of multiple interacting agents [6] that work in coordination with each other by communication and negotiation to achieve the ultimate goal, which are beyond their individual capabilities. In the context of power system, a MAS can be depicted as shown in Figure 1-1 below.

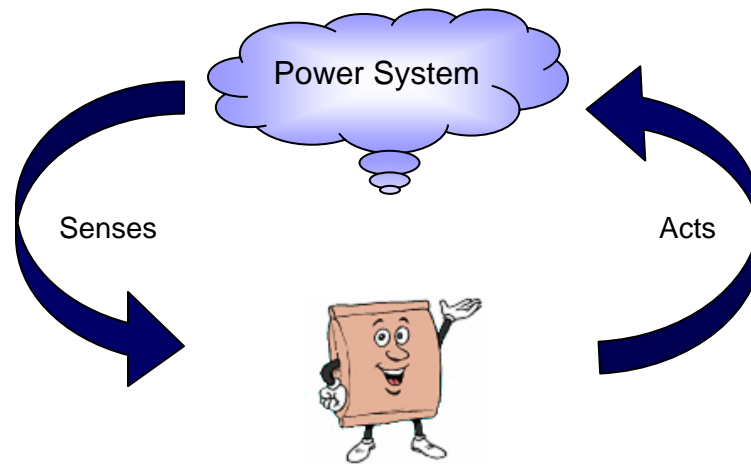


Figure 1-1: A Multi-agent System in the sense of Power System.

In addition, an agent has to be autonomous, which means that an agent can operate without direct human intervention. It can decide on its own, the required actions to be taken in order to achieve its objectives. Therefore, for MAS to be entirely autonomous, it has to have the ability of being an intelligent system. MAS without intelligence is no different to another software application, where the decision making process is designed by the programmer and if any unexpected situation other than a planned event occur, the system would crash. Hence, intelligent agents have to encompass the following characteristics [4].

Reactivity

Intelligent agents should have the ability to perceive the environment in which they live and respond in a timely manner to any changes or disturbances that take place in the environment in order to achieve their own goals

Proactiveness Intelligent agents should have the ability to take initiatives on their own in order to meet their goal directed behaviors

Social ability Intelligent agents should have the ability to interact with other agents in the system by coordinating, communicating and negotiating by using a standardized agent communication language

It is important to understand the difference between Agents, Objects and Expert Systems. The following points distinguish an agent from the traditional object [6].

- Agents embody a stronger notion of autonomy than objects. Agents decide on their own whether to take any action or not, when they receive a request from another agent
- Agents are capable of flexible (reactive, proactive, social) behaviors, but the object does not specify any behaviors
- MAS is inherently multi threaded, each agent has at least one thread of control

The major differences between an agent and an expert system can be listed as [6],

- Expert systems are not coupled to any environment in which they reside. They act through a user as a middleman
- Expert systems are not capable of reactive, proactive behavior
- Expert systems do not generally have social ability in the sense of cooperation, coordination and negotiation

1.3 Agent Communication

The previous section briefly described how agents differ from objects and expert systems. The major difference is the communication of agents among themselves. Hence, it is important to have a common Language for Agent Communication, without which, the coordination and negotiation will hardly be successful.

The communication protocol must be universally shared by all agents. It should be concise and have only a limited number of primitive communication acts [5]. The most popular Languages followed for Agent Communication are,

- Agent Communication Language (ACL) by the Foundation for Intelligent Physical Agents (FIPA)
- Knowledge Query and Manipulation Language (KQML)

For the agents to understand each other, they should not only speak the same language but also have to follow the same ontology. Two agents, while communicating with each other, need to agree on the same terminology. For example, in a power system, when two load agents are communicating with each other, when one requests for the other's 'flow', the recipient agent should understand what is meant by 'flow'. This is called the ontology. Ontology is thus a specification of a set of terms as follows.

“An ontology is a formal definition of a body of knowledge. The most typical type of ontology used in building agents involves a structural component. Essentially a taxonomy of class and subclass relations coupled with definitions of the relationships between these things” – Jim Hendler

1.3.1 FIPA - ACL

The Foundation for Intelligent Physical Agents (FIPA) is an organization, founded in 1996, to develop software standards for heterogeneous and interacting agents and agent based systems [7].

A FIPA ACL message will contain one or more message parameters, according to the requirements of the agent application. There is one essential parameter that has to be specified in all the agent communication, which is the '*performative*' parameter. In addition to this parameter, other important ones for effective communication are the '*sender*', '*receiver*' and the '*content*' parameters. FIPA ACL also allows the usage of user-defined parameters other than the ones which are specified by FIPA, showed in

Table 1-1 below. However, a prefatory string “X-“ must be used for the names of the user-defined message parameters which do not come under standard FIPA parameters.

Parameter	Category of Parameters
performative	Type of communicative acts
sender	Participant in communication
receiver	Participant in communication
reply-to	Participant in communication
content	Content of message
language	Description of Content
encoding	Description of Content
ontology	Description of Content
protocol	Control of conversation
conversation-id	Control of conversation
reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

Table 1-1: FIPA ACL Message Parameters [7]

This organization was dissolved in 2005 and an IEEE Standard Committee was set up in this place [8]. There are many agent systems, including Java Agent Development (JADE) Framework, which are using FIPA standards, at present.

1.3.2 Knowledge Query and Manipulation Language

KQML is another language and protocol which is used for communication between software agents and knowledge based systems. All the information required to understand the content of the communicated message is included in the communication itself. The basic protocol is defined by the following structure [5].

```
(KQML-performative
  : sender      <word>
  : receiver    <word>
  : language    <word>
  : ontology    <word>
  : content     <expression>
  ...)
```

At present, FIPA – ACL has superseded the KQML in most of the agent communications.

1.4 Problem Statement

This thesis will focus on the application of MAS for fault detection and reconfiguration and restoration. The study is based on a proto-type circuit, the Circuit of the Future (CoF) developed by Southern California Edison (SCE).

The algorithm ensures that the multi-agents, which are installed at all the nodes, sources, loads and switches, will communicate and co-ordinate with their neighboring agents, in order to provide a reliable power supply. This is achieved by re-routing the power flow when there is a line fault, and supplying the high priority loads, when there is a shortage of power supply and also switching on the DG when the need arises.

This study is based on the following assumptions.

- The distribution system is radial
- Reactive power sources (Capacitor banks) will maintain the voltage in the system, hence only the real power is taken into consideration
- Power factor of the system is 0.85
- DG is modeled as a backup source
- When a fault is identified, the protection (which is out of the scope of this work) is applied first and isolates the faulty lines, before reconfiguration
- Power losses in the distribution lines are not considered
- Loads are not intentionally disconnected by the customers or by any other electromechanical devices at the customer side

1.5 Approach

The original CoF, is simplified by lumping certain loads without affecting the switch locations and maintaining the original topology of the system. The circuit is modeled using Distributed Engineering Workstation (DEW), a power distribution software developed by Electrical Distribution Design Inc. The software used to design the MAS is

Java Agent Development (JADE) framework and Matlab. Fault detection algorithm is developed by JADE and reconfiguration algorithm is implemented in an enhanced Matlab agent model to allow conditional switching. These software will be presented in detail in Chapter 3.

Interfacing of these two software to perform a real time operation of agents in the power system is a challenging issue to address due to the complexity in different software platforms. Hence the agent application was implemented offline. Different fault scenarios were studied and the circuit was analyzed for better performance.

1.6 Thesis Outline

The outline of the remaining chapters is given in this section.

Chapter 2 will describe the literature overview of different multi-agent applications in power distribution system. It also covers a few other applications of multi-agents outside the power engineering field.

A detailed description of the software packages used in this thesis is given in Chapter 3. The advantages of these software and the suitability to the relevant applications are also discussed.

A mathematical model, based on graph theory, for fault detection and reconfiguration algorithms will be presented in Chapter 4. This chapter will include the modeling of the system and the Circuit of the Future that is used in this work to perform the simulations.

Chapter 5 will present the simulations and results of the model. Finally, conclusion of this study and the future work is detailed in Chapter 6.

Chapter 2

LITERATURE REVIEW

This chapter presents a comprehensive overview of the previous research work carried out on multi-agents and its applications in the area of power distribution system. The section starts with the discussion of the advantages of MAS and the limitations posed in implementing the system in the real world. This is followed by a literature review of MAS applications in the areas of voltage control, protection coordination, distributed generation, and specifically for fault detection and reconfiguration and restoration. The last part of the chapter covers the technical challenges and few other applications of MAS in areas other than power systems.

2.1 General Overview of MAS in Power System

Multi agent system have been applied to several areas in power systems, such as reconfiguration and restoration, fault detection, protection coordination, voltage stability control, reactive power control, electricity market pricing etc. These theoretical based researches are being applied, at present, to terrestrial power systems to test the reliability of MAS. Recently more research work based on complicated applications in the real world is being carried out.

In their paper, Cartes and Srivastava analyze the potential of agent application and their future in power industry [9]. They present a SWOT Analysis (Strengths, Weaknesses, Opportunities and Threats) framework for agent based applications in the area of power systems. The SWOT matrix they presented is given below in Figure 2-1.

<p>Strengths</p> <ul style="list-style-type: none"> • No Single point of failure • Decentralized data processing <ul style="list-style-type: none"> - Fast Operation - Task Distribution • Ease of tackling complex problems • Natural Solution • Leads to scalable, modular and flexible solutions 	<p>Opportunities</p> <ul style="list-style-type: none"> • Technological Advances • Collaboration/Cooperation • Develop generic agents/MAS • Existing hardware/software infrastructure • Learn from other distributed applications
<p>Weaknesses</p> <ul style="list-style-type: none"> • Can lead to unstable operation • Complexity • Lack of understanding • Lack of Experts • Still in its infancy • Lack of practical applications in power area 	<p>Threats</p> <ul style="list-style-type: none"> • Unimagined problems • Acceptability or concept still not taken seriously • Centralized schemes • Security Issues

Figure 2-1: SWOT matrix for current Agent Applications to power systems [9].

The paper discusses in detail the Strengths, Weaknesses, Opportunities and Threats of applications of MAS in power engineering and also about the current applications of multi-agents in power systems. It also proposes a new broader structure for future MAS application for power systems, which caters the entire control operation of power systems.

In [10], the authors stress the fact that MAS is the solution for many issues in today's deregulated power systems. The paper identifies Distributed computing, communications and data integration as the three main problems that affect the power system. The authors point out that the characteristics of MAS, such as modularity, extensibility, flexibility and the integrated approach can tackle the three issues mentioned before. Their group has a four step methodology for MAS construction: analysis, design, implementation and deployment. Starting with the analysis, the problem definition and the objectives are set. In the second stage, definitions of each

agent's role are developed and the interactions between the designed agents are identified. The third step is where the selection of appropriate agent platform and architectures are considered. And finally the agent model is deployed and tested. This is a very methodical framework to design an agent application.

2.2 MAS in Automatic Fault Detection

The concern of power system reliability very much depends upon continuous monitoring of the system, fault detection and reconfiguration to supply maximum load as possible during a fault or disturbance in the system.

The paper by Li Liu et al. [11] describes the need for fault detection in naval shipboard power system. The paper discusses the practical fault detection and diagnosis problem along with prognostics from control engineering perspective. In their fault diagnostic framework, the task has been defined as to constantly monitor the process and from the available observations, to identify an indication to decide whether there is a fault or not, and to identify the fault location.

Different approaches are followed for dynamic fault detection process, such as modeling and estimation method [12], [13], using analytical redundancy [14], [15], factorization approaches [16], etc. and more recently neural network, artificial intelligence and extensive numerical methods have been applied. A paper was published in 2004, summarizing the efforts made towards fault detection [17]. In this paper, the authors have described the methods followed towards this research area.

Recently, Huang et al. have presented the multi agent approach for fault detection. The fault is detected by using nonlinear parameter identification techniques. Once the fault is detected, the diagnosis agent makes further decisions on the fault mode, fault location and fault severity. This study has been performed by using particle swarm optimization approach [18].

2.3 Reconfiguration & Restoration by MAS

Many authors have conducted research in the area of reconfiguration and restoration of power system, primarily in the shipboard power system. Solanki et al., have presented the restoration of power system by distributed reconfiguration [19]. In their multi-agent model, they have introduced three types of agents, named Switch Agents (SAs), Substation Breaker Agents (SBAs) and Tie Breaker Agents (TBAs). The objective in this work is to detect and isolate the fault and supply as much load as possible, while assuming that the components are not overloaded and radial configuration is maintained. Fault detection is based on monitoring the fault currents of neighboring agents. The power system model was developed using Virtual Test Bed (VTB) to simulate the P-Q model of the three phase power system and the agent model was developed in Matlab.

In a later work [20], the same authors presented a better model for the restoration problem. This model addresses the load priority, limitations on generating source capacity and cable transfer capacity. However, this paper also assumes that the configuration is radial. The agent model was developed in JADE with Switch Agents (SAs), Load Agents (LAs) and Generator Agents (GAs) and the power flow is calculated by forward and backward propagation. These agents communicate only with their neighboring agents. The interface between JADE and VTB had been developed by Florida State University for this system.

Gomez-Gualdrón et al., in their paper [21] have presented another approach for reconfiguration of electric power distribution system. In their work, the power system simulation is done using Matlab-Simulink and JADE was used to develop the agent model. The major concentration of this paper is given to the integration of these two software models, i.e. the communication middleware. The importance of the middleware is to let the MAS sense when a fault occurs in the power system and to make necessary control actions. The communication middleware allows the JADE MAS to send and receive data to and from the Simulink power system model. This has been achieved by using Transmission Control Protocol (TCP) sockets. This model consists of a Server Agent, Zone Agent and a Communication Agent. The same authors, in their paper [22], have modeled the MAS as an Intelligent Power Router (IPR), where the intelligence of

IPR is provided by the software agents. These power routers are deployed in the power distribution network for reconfiguration and re-routing the power flow the same way as routers do in the information distribution network. Based on their model requirements, FIPA Contract Net Interaction Protocol is used. In the Contract Net Interaction protocol, an initiator agent corresponds with the responders in order to perform its tasks. The responders may send a proposal or refusal according to their system conditions at that time. The negotiation continues with the responders who send proposals. After receiving all the proposals, the initiator selects the best proposal according to its criteria and informs the relevant responder with the acceptance message. Those responses that are not selected will be informed with a rejection message by the initiator. Then the respective responder will perform its intended task in order to achieve the overall objective of the system.

The work discussed above for multi-agent reconfiguration of power system assumes that the system topology is radial at all times. The paper by Huang et al. [23] attempted to address the mesh power network. In this method, there are several rounds of negotiations along the ring configuration addressing the source generation capacity and the load demands. The series of accumulation of messages during multiple negotiation rounds causes Redundant Information Accumulation (RIA), leading to an unstable information flow. The authors propose a reconfiguration method to avoid RIA where the algorithm generates a tree structure to break the existing rings. The simulation was done using JADE as the agent platform and Real Time Digital Simulators (RTDS) as the power system modeling tool. The agents developed in JADE have been implemented in iPAQs, which are pocket Personal Computers, and communicate with RTDS through a Field Programmable Gate Array (FPGA) interface.

Takeshi Nagata et al. have immensely contributed on the application of multi-agent system for power system restoration. He initially addressed the restoration issue by the usage of expert system and mathematical programming [24]. In this approach, the restoration is targeted as an integration of Mathematical Programming with Expert System. The objective function is solved efficiently by decomposition-coordination strategy. He solves the restoration problem by using a two step process. Firstly, the target configuration is determined and then sequence of control operations of the switches is determined based on the target configuration.

Later the same author proposed an autonomous distributed multi-agent model for power system restoration in [25], [26] and [27]. Starting with [25], he introduced Bus Agents (BAGS) in the system to find the optimal target network configuration. In [26], he introduced a single Facilitator Agent (FAG) to act as a manager in the decision making process in addition to the BAGs. In his later research [27], he developed a system that not only solves the problem of minimizing the de-energized loads, but also ensures that each agent solves its own bus's problem autonomously. These are achieved by two kinds of agents in the system, namely, Bus Agents (BAGs), relevant to the substation bus concerned, and Junction Agents (JAGs), corresponding to the junction of a multi-terminal transmission line. The purpose of the BAG is to restore the load directly from its own bus. The JAGs operate to control the cooperation and competition between BAGs connected through a junction. Both the BAGs and the JAGs have their own strategies for restoration. Message communication between agents is achieved by general-purpose Agent Communication Language (ACL) which is mounted on Extensible Markup Language (XML). This method also assumes that the configuration is radial.

In [28] Nagata proposed a new decentralized multi-agent approach for the bulk power system restoration to enhance the speed of autonomous decision making process. The system is constructed with two-level hierarchical architecture. In the upper level, several Local-area Management Agents (LMAs) and Remote-area Management Agents (RMAs) are implemented while several Load Agents (LAGs) and Generator Agents (GAGs) are located at the lower level. The LMAs have two objectives: to restore within their local areas and to interact with other LMAs in order to achieve the overall objective. A RMA has only one objective, which is to assist the restoration of the loads within the remote area. This has to be achieved by series of negotiations between the RMA and the neighboring LMAs. LAGs and GAGs correspond to load management system and generator management system which store their respective loads' and generators' local information. In addition to their local communication, they also interact with the upper level agents.

Apart from these, Nagata et al. have also proposed several agent methodologies to address the issues in Bulk Power system restoration. In [29], the authors have introduced a single Independent System Operator Agent, several Local Area

Management Agents (LMAs), Local Facilitator Agents, Generator Facilitator Agents, Remote Load Facilitator Agents, Load Agents and Generator Agents to address the entire bulk power system.

Contribution towards bulk power restorations have also been made by Dong Liu et al. In [30], the authors proposed a hierarchical architecture for MAS bulk power restoration. There are two major agents, Management Agent (MGAG) and Practical Components Agents (PCAG) in a Power System. PCAGs constitute of several Generator Agents, Substation Agents and Load Agents. MGAG is generated for a particular system to be in charge of the subsystems and administer the negotiations between the other agents to check for constraint violations and to facilitate path search for restoration.

2.4 MAS in Distribution System with DER

Distributed Energy Resources have penetrated greatly into the power transmission and distribution systems mainly due to deregulation of power industry. Installation of DERs in the system needs high consideration which makes the system to be very complicated to be reconfigured. In addition, it impacts on the protection issues, which will be discussed later in 2.5.

A dynamic hybrid multi-agent paradigm to achieve scalability for control, reliability and higher power quality of a large network of power generation including distributed energy resources, transmission, load and compensation sources is presented in [31]. In the example, agents are developed for system stability and harmonic and reactive current compensation. In this approach, the authors emphasize that the scalability is a significant issue due to the size of the power system, which can vary from thousands to tens of thousands of nodes with an array of interconnections between the nodes. A hybrid structure of agents is proposed where the peer agents communicate and collaborate with each other and they dynamically select a leader to communicate with their parents. In their model, compensator agents are developed for power electronic compensators that provide a range of service such as reactive power generation, power flow control, harmonic compensation, voltage regulation or dynamic control over the frequency and voltage. These agents will be integrated with generating sources and

loads. A vast set of decision criteria is needed for the agents in order to make the ultimate decision to enhance the reliability and power quality of the power grid.

Hiyama et al. worked on the application of MAS for control and operational issues in distributed system with Distributed Energy Resources [32]. In their work, Energy Capacitor System (ECS) is used as a storage device and plays the main role of load following operation. A hierarchical multi-agent based operation and control too has been used in the work. This study was later expanded to cater for power system reconfiguration with DER. Accordingly, another paper has been presented recently where, the authors describe an approach to address the DG issues in reconfiguration [33]. This is a first step towards the reconfiguration of the system, which comprises of DGs and capacitor banks.

2.5 Protection Coordination

Many authors have presented their work in the area of multi-agent system application for protection coordination, with and without considering Distributed Energy Resources in the system. The application of agent technology in protection relays started in mid 1990s, and was mainly used for designing of protection systems. Coury et al. [34] proposed an agent based current differential relay which can be used in the existing communication network for power system. Relays play a vital role in power system operations with very low tolerance for failure. Hence, MAS based relay coordination can be very promising against the conventional methods for relay coordination. The additional features of the agent based relay compared to that of the conventional relays are that, the agents constantly exchange information about the status of the system and regarding failure of any equipment, the agent would detect equipment failure based on measurements, and send trip signals to relevant breakers in order to isolate the faults and thirdly, agents are also be able to detect communication channel failures [34].

In [35], the authors have developed a MAS named Protection Engineering Diagnostic Agents (PEDA) which integrates the Supervisory Control and Data Acquisition (SCADA) with Digital Fault Recorders (DFR) to perform post-fault disturbance diagnostic. The development of PEDA has reduced protection engineers' involvement in fault diagnosing and gives a timely and more accurate solution for the number crunching problem. PEDA

also has the capability of scalability and flexibility for future enhancements. Following this research initiatives, the work carried out by South China University of Technology [36], introduced a step by step approach to achieve the protection relaying using PEDDA. Firstly the information gathered from SCADA is stored in a database, and then the users start the operation of the agents. Each agent acquires information from MAS and initializes itself. The other agents in the system identify the initiator agent and perform various parameter checking and select the agents to take the initiative for protection of power network and also inform back to the overall MAS. The MAS decides if protection should function or not and if it has to function, which protection operator has to execute and accomplishes its results by cooperating with the other agents in the system.

A framework of supervisory control of discrete event system is applied in [37] for agent based power protection scheme. This control system uses feedback information to achieve the objective function. A design of a supervisor agent is presented in detail in this work. The task of the supervisor agent is to coordinate the behavior of relay agents in order to isolate the fault areas by minimizing the circuit breaker operations.

In a system where Distributed Generators are in operation, the protection scheme has to be changed vastly to cater for DG functioning. The traditional protection schemes with radial networks will not be in a position to cater the high impedance faults where a DG is present. In [38], a digital relay is designed as a relay agent, which searches for information from the other agents in the system by interacting with them and perform protection task to accomplish its goals.

The authors of [39], presented a protection coordination mechanism for MAS. In the proposed architecture, there are three types of agents. A DG Agent, which communicates with the relay agents to provide information regarding the DG status, a Relay Agent, which is implemented in each of the relays in the system, to detect relay malfunction and breaker failures, and an Equipment Agent to collect local information about the other equipments in the system. This agent based relay coordination has the ability to self-check, self-correct and rapidly act during fault conditions.

The same authors presented the application of agent technology to substation protection co-ordination in [40]. In this paper, in addition to the relay settings, communication

between agents also has been investigated in detail. In the proposed architecture, a Substation Management Agent, which plays the role of an interface between the inside and outside of the substation, has been introduced in addition to the other agents in the system, discussed in the previous paragraph.

Apart from the work discussed above, there is also contribution towards mobile agents being applied to power system protection and control [41], [42], [43].

2.6 Voltage Control

Voltage regulation and reactive power management is a major concern in power distribution system. Application of MAS to regulate the voltage is a very promising area of research as MAS can autonomously monitor and control the voltage issues in a distributed manner and give solution to the system voltage issues within very short time spans, which enhances power quality. A few researchers have contributed towards this area. In [44], the author mainly addresses the secondary voltage control issue with three different power system voltage controllers, Automatic Voltage Regulators (AVR), Static Var Compensators (SVC) and Static Compensators (STATCOM) for power system voltage management in system contingencies. This system is implemented based on MAS. In the method proposed, there are three agents in the system, namely, AVR Agent, SVC Agent and STATCOM Agent. The communication between the agents is the type of 'request and response'. Each agent will be monitoring the voltage at its end and when it finds out that there exists a voltage violation, it tries to resolve the problem by activating its reactive power reserve. If it fails, then that particular agent will request from the other agents in the system to solve the issue. The agents, which receive the request, will respond with a proposal, after checking that the intended proposal does not conflict with its own individual objectives.

Later in 2005, Gehao et al. also addressed the same issue of secondary voltage control scheme. In normal operating conditions, the MAS based control system acts as a conventional secondary voltage control system, acting towards optimal Var/Voltage control. During contingencies, a contract net protocol, used in MAS applications, is introduced to achieve coordination between voltage control agents to remove voltage violations [45]. In the model, the regional secondary controller is considered as a

Coordination Agent (CA) and the other voltage controllers, such as AVRs, SVCs, synchronous condensers, static condensers etc are managed by the Execution Agents (EA). During contingencies, the MAS switch from the normal mode to the emergency mode. In this mode, the relevant EA will make its own decision to change the setting of the primary voltage controller for rapid restoration of voltage. If an EA cannot resolve the problem, it has to send a request message for assistance to CA. Subsequently the CA will coordinate with the other EAs to bring back the voltage to the desirable limits.

2.7 Technical Challenges

As discussed above, multi-agent technology is being applied to many applications in the area of power systems. This technology is moving from the research arena to the utilities for harnessing the benefits of the MAS.

Despite the growing awareness and the research of the multi-agent technology in the power engineering area, there are some fundamental questions, from researchers and in particular the industrial partners, posed related to the benefits of MAS, such as, how it differs from the existing approaches and what are the applications etc. [46]. In their paper, the authors clearly discuss that communication between the agents in a MAS environment is the major fact which differentiates a MAS to the other systems that are currently in use. Power system applications require more flexible and extensible systems, which should be able to handle enormous amount of data. The paper considers the fact that MAS is a framework for integrating different Artificial Intelligence Techniques. Hence, in developing agents, the other intelligent systems, such as Expert systems, Artificial Neural Network etc can be used. The authors also put forward the technical challenges faced by the MAS applications. They can be briefly categorized as,

- Proper selection of Platforms
- Need of toolkits for reuse of existing agent behaviors
- Lack of guidance for new researchers on intelligent agent design
- Following the Agent Communication Languages and Ontologies
- Managing security Issues
- Need for Agent mobility
- Lack of experience in the area

- Need to report the efforts by the industry, whether the effort is a success or failure

The same working group's second part of the same paper presents the design and implementation issues of MAS. They emphasize the fact that the design should be developed by following the MAS standards such as the IEC Standard on Common Information Model (CIM) and also to design in such a way that it allows the interoperability between agents, which are developed by different designers. The authors recognize the problem being that different power system applications have been developed by different application specific ontologies, which is simply given as the 'agents speak the same language but do not share a common vocabulary' [47]. Solution for this issue, as presented in [7], is to create an ontology agent, which provides ontology related services.

Moreover, it is another challenging task to integrate the multi-agent applications to the power engineering applications. Morejon et al. propose a methodology to interface the MAS application, JADE to the power system application, which is developed in Virtual Test Bed (VTB). The authors have implemented this interface through a communication mechanism, Common Object Request Broker Architecture (CORBA). CORBA allows application interoperability, which does not depend on application platform, operating system or programming language [48]. It is an open distributed object computing infrastructure being standardized by the Object Management Group (OMG) [49]. It automates many common network programming. This mechanism enables the formation of gateways between external applications and simulation in VTB. The agents which are on each element in the power system can receive and send signals during runtime.

2.8 Other MAS Applications

Multi-agent system is a modern, decentralized, autonomous and intelligent approach for many areas in the field of engineering. Hence, a system with a huge network design can be easily automated and controlled by exploiting the MAS. Many areas of engineering have already moved in the direction of MAS technology. A few of them are discussed in brief below.

2.8.1 Highway Traffic Management

Traffic congestion during peak hours causes less productivity, wastage of time, increased number of accidents which risks the public health, increase in air pollution by vehicle emissions etc. Hence, it is prudent to design an autonomous traffic management system, which mitigates all the above mentioned effects. In [50], the authors present a study to address the issues of congestion in Automated Highway System (AHS) that targets an efficient reallocation of network capacity over time and space to ensure efficient routing, departure and arrival time for any vehicle. The model has five agents; the General Agent is for communication and forms an interface between the human controller and the other agents in the system; a Zone Agent, interacts with Section Agents, Truck Agents and the General Agent, and it allows to manage works, weather related issues, accidents, incidents etc. on the whole highway of the particular zone. It proposes the shortest itinerary to follow to the section agent; a Section Agent knows the traffic status in its section. The main function of this agent is to regulate the traffic with the information it has from the other agents; a truck agent represents the automobile together with its driver, and has all the needed information regarding both the vehicle and its operator. Finally, the Security Agent's duty is to manage the incidents and faults. This agent is responsible for reconfiguration of the system to re-route the traffic flow and to inform this to the relevant truck agents. This aggregated behavior manages the highway traffic with very less human intervention.

2.8.2 Agent Based Air Traffic Control

Due to the increasing intensity of air traffic and requirement for safety issues, the autonomous MAS have shown a promising solution for the Air Traffic Control (ATC). The authors of [51] present a system which contains several agents playing the role of assistant aircraft crews and assistant air traffic control operator. In the presented model, air space is categorized into arrival and approach zones. The objective of the model is to minimize the human intervention in air traffic management by delegating humans' functions to pilot assisting agents. These agents coordinate the airline movement in the arrival zone, by predicting air traffic configurations, based on air space topology and safety policies. In the approach zone, the assistant air traffic control operator agent is

responsible for air traffic control. This mechanism can very largely reduce the number of human operators.

2.8.3 Mine Detection

Another promising application of MAS is the searching for items or resources in unknown environments [52]. Zafar et al., in their research work, studied a hybrid architecture to implement mine detection, obstacle avoidance and route planning by using MAS. There are different techniques to design the mine exploration. The authors have followed frontier based exploration. This makes an agent to construct its own map and explore the entire area. This mechanism also deals with post exploration route planning to generate a safe route for given target destinations.

There are many other applications, which are investigated in the research world, such as e-commerce, auto-robot, assembly line etc. In automobile robot application, the MAS will decide on the driving actions according to the changing environments and make reaction to emergency conditions [53].

2.9 Conclusion

A comprehensive review of literature, related to MAS in power system applications, is presented in this chapter. It concluded with a brief discussion of MAS applications in other engineering disciplines.

Chapter 3

SOFTWARE PACKAGES

The application of MAS in power engineering moved from the research arena into the stage of implementation on proto-type circuits only very recently. Hence, the research work related to MAS greatly depends on software simulations. Thus, it is very important to choose suitable software packages that can cater for the application specific needs.

The software requirement for this project can be explained clearly from Figure 3-1. Power system software and MAS development software are in two different layers and there needs to be an interface between them for the MAS to act on the Power system.

Accordingly, several options were analyzed for both power distribution simulation and for agent simulation and to interface both the applications. Finally, Matlab, Distributed Engineering Workstation (DEW) and Java Agent Development Framework (JADE) are selected to model the simulations in this research work.

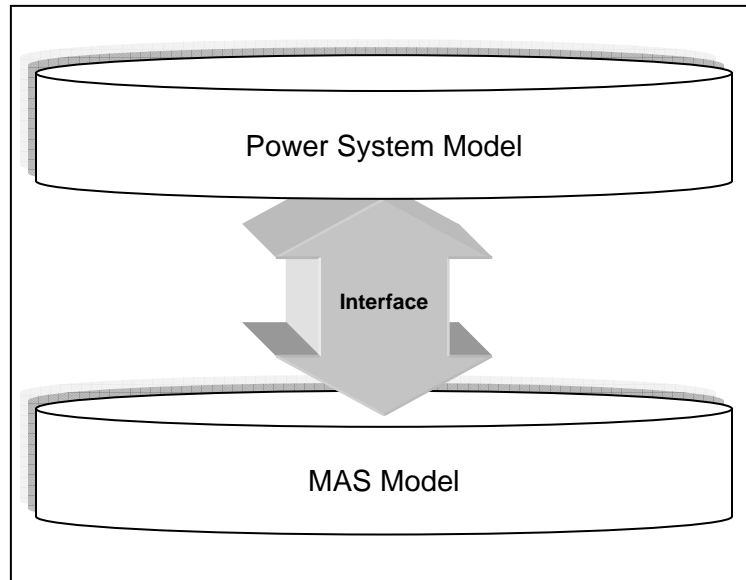


Figure 3-1: Architecture of the Simulation Model.

3.1 Matlab®

Matlab®, the Matrix Laboratory, is a high-level language and interactive environment for algorithm development, data visualization, data analysis and numeric computing. Matlab enables to perform computationally intensive tasks faster than the traditional programming languages such as C, C++ and FORTRAN [54]. It supports matrix and vector operations, which are key to power engineering. In addition, it facilitates fast and easy development and execution of matrices and other computations.

Matlab is a very powerful tool that is faster to program; as the basic administrative tasks such as variable declaration, memory allocation, data type specification etc. do not need to be performed. Hence, it is faster and easier to use. The Object Oriented Programming (OOP) is one of the best features in Matlab. OOP improves the ability to manage software complexity, mainly when developing and maintaining large applications and data structures. In addition, it has features to provide the ability to visualize the engineering solutions and to create Graphical User Interface (GUI) for users to perform interactive tasks.

In this work, algorithms are developed through object oriented programming for reconfiguration and guarded statements are used to make it specific for restoration of the Circuit of the Future. A GUI is also developed for easy user interaction.

3.2 Distributed Engineering Workstation© (DEW)

3.2.1 Description of DEW

DEW is a software package for power distribution system that provides an open architecture environment with integrated data and applications, developed by Electrical Distribution Design Inc. The applications that are available with DEW vendors are: power flow, load estimation, line impedance, fault analysis, capacitor placement, phase balancing, economic analysis, reconfiguration and restoration, overcurrent coordination etc. These applications come individually so that the users can purchase only the required applications for their individual needs.

DEW consists of a comprehensive database and schematic editor for the distribution system model. Analysis, design and application modules access database and exchange data through common functions. The open architecture allows external application modules to be added to the workstation by any user [55].

The figure shown below shows the main screen of the DEW's graphical user interface. It has a component list on the right, from which, the power system components can be easily dragged and dropped to the workspace. Voltage profile of the highlighted load at the end of the middle feeder, plotted against the distance is also shown in the diagram.

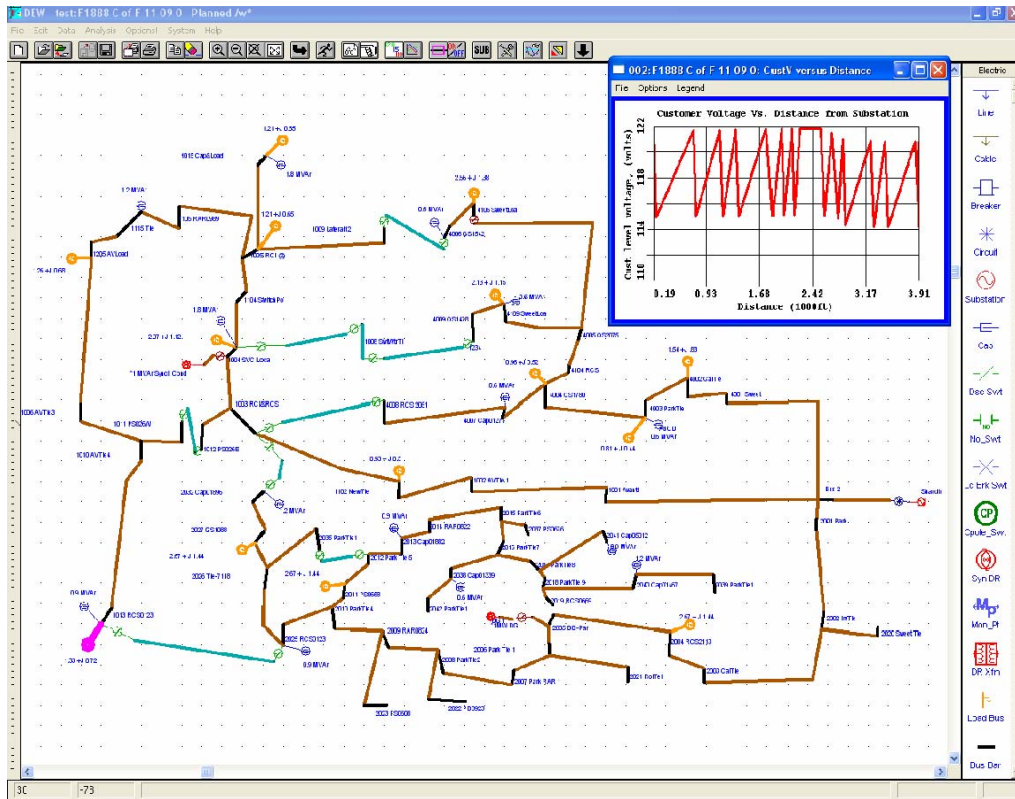


Figure 3-2: Snapshot of DEW Working Environment.

Presently though the utilities have enormous amount of data for distribution studies, these data are neither integrated nor in the same location, which is a major drawback for any power system analysis. This was the main objective in the development of DEW. This software package has a data schema design, which takes care of the vast data in a central location and can be accessed through any applications in DEW. This integrated nature allows all the applications to share the same database [56]. The data schema consists of a relational database, which is in the hard disk and in memory. During initialization of DEW, instead of reading all the data from the database, only the most commonly used data are loaded into the memory, this saves a lot of time. The remaining data can be accessed during the performance of any specific applications.

The major components of DEW are the Executive, database, GUI, Application Programmer Interface (API) and application modules. The DEW Executive manages the operation and provides the API, GUI and database interface for the application modules [56]. The API integrates the application modules into the framework. All the applications

use GUI for user interaction and display. As illustrated in the following figure, it can be seen that the data from Supervisory Control and Data Acquisition (SCADA), Corporate Data and other data such as financial, statistical, geographical, operations data are utilized for the application via an interface, Database Access Integration Services (DAIS) [57].

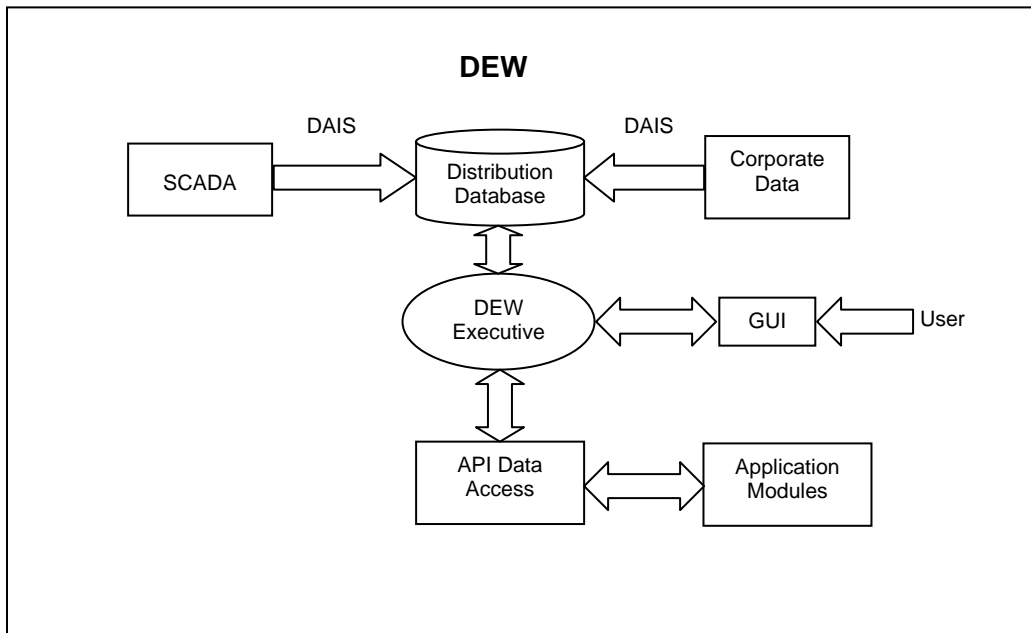


Figure 3-3: Major Functional Parts of DEW [57].

DEW is easy to work with, as it needs only a personal computer with Microsoft Windows, which is used by larger number of users. The programming languages used to develop DEW are C and C++. Application developers can use these languages to create dynamic link libraries to program new applications.

3.2.2 DEW Modeling Features

DEW has a lot of features available for modeling the distribution system. Loads can be modeled as spot loads, special loads, time varying loads etc. It also allows to model annual percentage growth of the distribution feeders. Load priority also can be given while designing.

In the distribution system, both radial and loop connections can be made by switching on/off the switching components, such as breakers, switches, disconnectors etc. When it detects that there is a loop, the network power flow is automatically run, instead of normal power flow. In the power flow analysis, the user can select it to run either in constant power, constant current or voltage dependent load models.

The capacitor banks can be modeled as switched or multiple step controller, the controlled variable being, voltage, power factor, reactive power flow, time control, current control etc. There is also the possibility to model co-generators which allows studying the distributed generation.

The output of the results can be either displayed in the screen or plotted against time/distance or written into reports in the form of text files.

3.2.3 Pros and Cons of modeling with DEW in this work

In this work, several power system software packages were analyzed for power flow calculations. The requirement that this work had was that the power system software needed to have the ability to be interfaced with JADE or Java platform or Matlab. There are different ways of transferring data from one to the other. However, it was not possible to constantly transfer data between both the packages. The package that was selected by the project was DEW. This software has many capabilities; especially the main reason which attracted DEW to the project was that it had the capability of handling three phase power distribution system. So that, unbalanced power flow can be analyzed with ease, since the power distribution system is usually unbalanced, unlike the transmission system.

The analyses that were made available for WVU were: power flow, network fault analysis, load estimation and DR control. The need of interfacing DEW with another software was necessary to handle the intended application of MAS. The information that was available at present was not adequate to handle the interfacing issue.

Because of the above mentioned challenges, DEW was used only to run the power flow of the CoF and the results were written to a text file, which is accessed by JADE for agent application.

3.3 Java Agent Development Framework

JADE is a software framework fully developed in Java language, which facilitates implementation of MAS through a middleware. It is developed in accordance with FIPA standards discussed in section 1.3.1. In addition, it has graphical tools which assist in development stages and to visualize the agent communication.

It is the most widespread agent-oriented middleware in use today. JADE was initially developed in 1998 by the Research and Development department of Telecom Italia s.p.a., but in 2000, it became open source as a community project and was distributed as open source under the LGPL (Library Gnu Public License). JADE can be freely downloaded from Telecom Italia website [7], with documentation, examples and other important information.

JADE allows the agent platforms to be distributed among several machines [58] according to FIPA standards. The FIPA Agent management model, which has different platforms, is shown below in Figure 3-4. Where, AMS is Agent Management System and DF is Directory Facilitator.

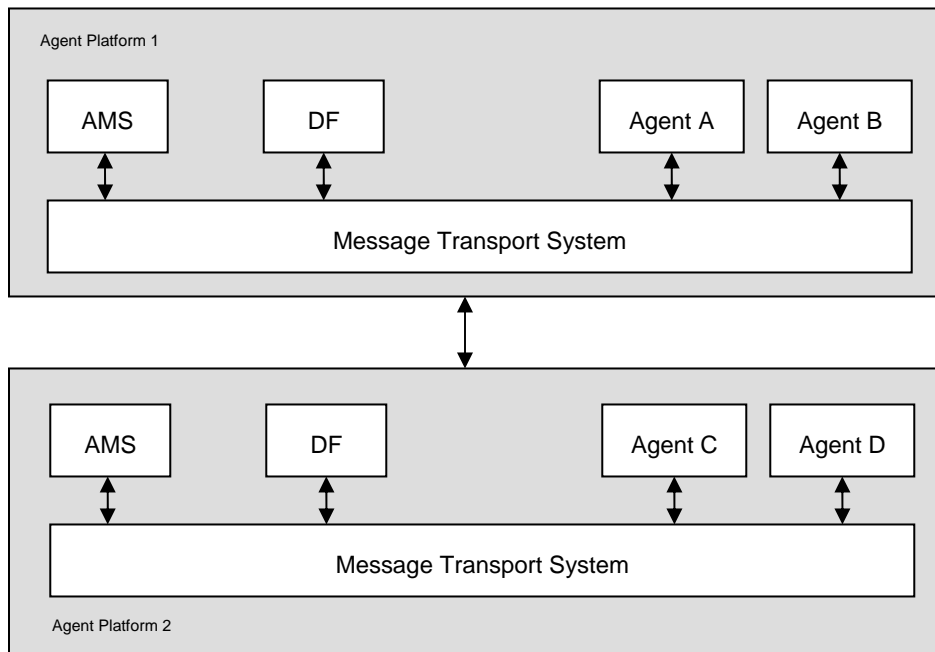


Figure 3-4: FIPA Agent Management Reference Model.

3.3.1 JADE Architecture

A JADE platform is composed of different agent containers that can be in different computers and are distributed throughout the network. These containers provide all the agents that are in them, with all the services needed for accommodating and executing them. There is a special container, called the main container, which represents the bootstrap point of a platform. This is the first container to be launched and all other containers must join the main container by registering themselves [59].

Two special agents are automatically instantiated and started by JADE, when the main container is launched. They are:

- The Agent Management System (AMS)
This is the agent, which is responsible for supervising the entire platform by acting as the contact point for all agents which need to access the white pages and to manage their life cycles. Every agent that is living in the platform is automatically registered by JADE to the AMS in order to get a valid Agent Identification (AID)
- The Directory Facilitator (DF)
The DF facilitates with the registering of the services of individual agents by implementing the yellow page services. DF also caters the requirements made by different agent to inform them when a facility becomes available.

Apart from the above two agents, there are a few other agents that facilitate monitoring and administration of agent management. They are,

- The Remote Monitoring Agent (RMA)
RMA is a tool that implements the GUI. This provides a visual interface to monitor and administer a distributed JADE platform. At initialization, RMA subscribes to AMS to notify all the platform-level events.

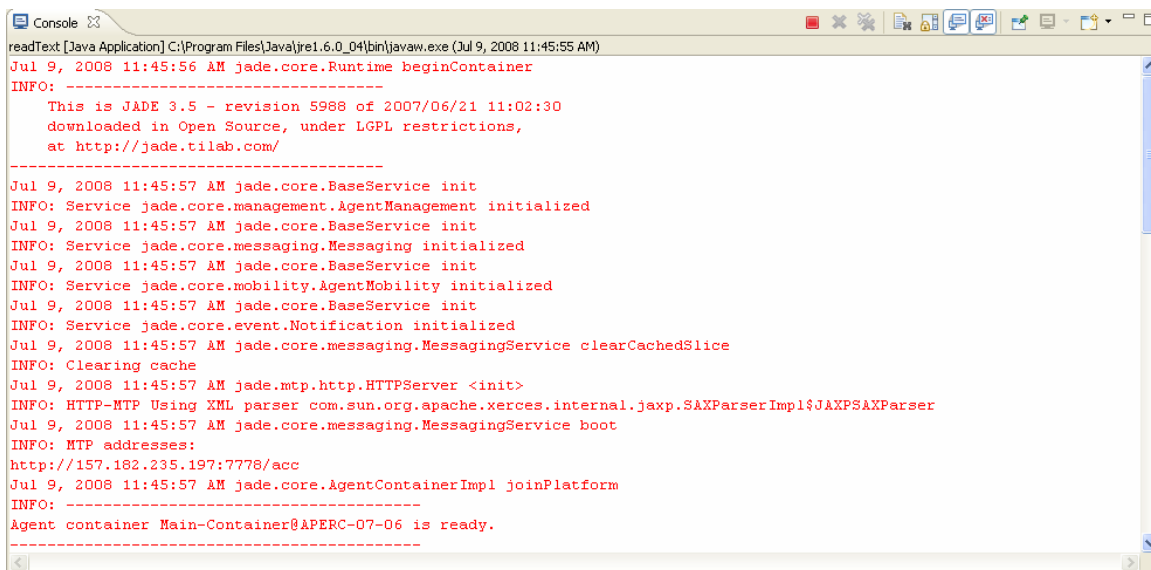
- **The Dummy Agent**

The dummy agent is capable of sending and receiving custom messages that can be loaded from and saved to a file by using a GUI. This is useful during programming stages to test the performance of application agents.
- **The Sniffer Agent**

The Sniffer agent is very useful to monitor the message passing between agents in the platform. The conversations among the agents are clearly shown in a GUI, where the conversations are indicated by arrows and separate conversations are distinguished with different colors.
- **The Introspector Agent**

The introspector agent is used to debug the behavior of a single agent and monitors the life cycle and communications of that agent. It also introspects the behaviors of an agent, such as, which behaviors are executed, which were passed through without execution and how it reacts for incoming messages.

A typical output of the JADE run-time environment and initialization procedures when launching JADE is shown in Figure 3-5 below.



```
readText [Java Application] C:\Program Files\Java\jre1.6.0_04\bin\javaw.exe (Jul 9, 2008 11:45:55 AM)
Jul 9, 2008 11:45:56 AM jade.core.Runtime beginContainer
INFO: -----
This is JADE 3.5 - revision 5988 of 2007/06/21 11:02:30
downloaded in Open Source, under LGPL restrictions,
at http://jade.tilab.com/
-----
Jul 9, 2008 11:45:57 AM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Jul 9, 2008 11:45:57 AM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Jul 9, 2008 11:45:57 AM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Jul 9, 2008 11:45:57 AM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Jul 9, 2008 11:45:57 AM jade.core.messaging.MessagingService clearCachedSlice
INFO: Clearing cache
Jul 9, 2008 11:45:57 AM jade.mtp.http.HTTPServer <init>
INFO: HTTP-MTP Using XML parser com.sun.org.apache.xerces.internal.jaxp.SAXParserImpl$JAXPSAXParser
Jul 9, 2008 11:45:57 AM jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://157.182.235.197:7778/acc
Jul 9, 2008 11:45:57 AM jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@APERC-07-06 is ready.
-----
```

Figure 3-5: Snap shot of JADE run-time environment.

The following figure depicts a typical GUI of the JADE platform.

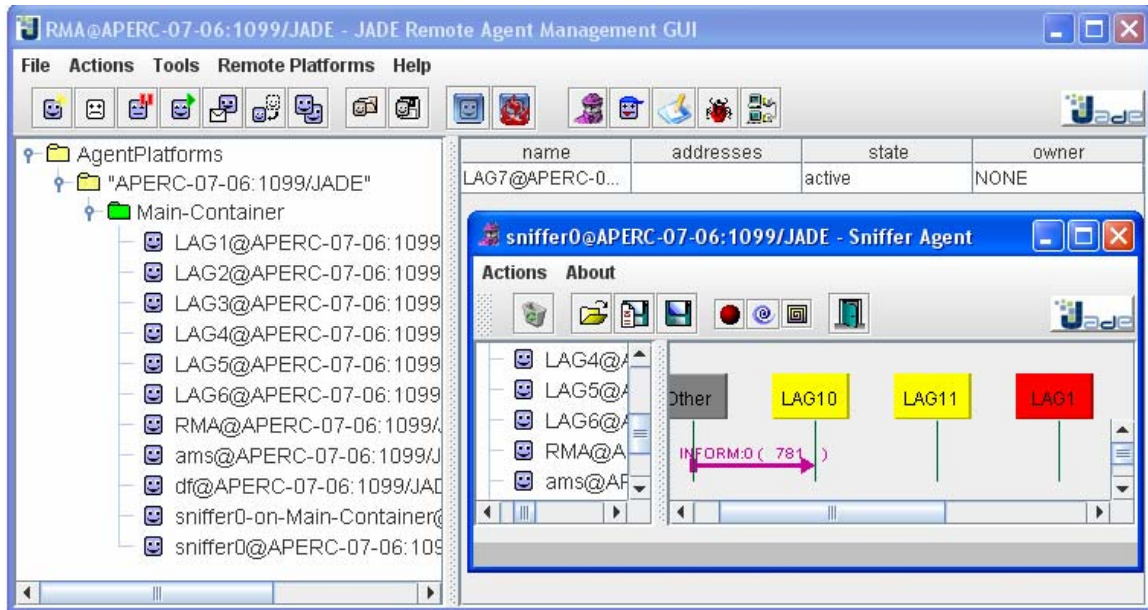


Figure 3-6: Snap shot of JADE GUI.

JADE software is freely downloadable from [58]. Before launching the JADE platform, it is necessary to set the local java class path, which is necessary in order to compile the agent classes. The important libraries that have to be launched prior to any JADE application are `jade.jar`, `jadeTools.jar`, `http.jar` and `iop.jar`.

3.3.2 JADE Programming

Creating a JADE agent is, by building a java class that extends `jade.core.Agent` class and implementing the `setup()` method. This method is where the agents are initialized. The intended performance of an agent is implemented within its *behaviors*. An agent can perform three types of behaviors, one-shot behaviors, cyclic behaviors and generic behaviors.

Each agent instance has a unique *agent identifier*. The `getAID()` method of the `Agent` class retrieves the local identification of the particular agent. An agent will also have a globally unique name which includes the agent name and the platform in which it is generated from. These AIDs are mostly used in the message passing stages.

Agent communication is the most important feature in MAS. Each agent that is created has a mailbox where the messages are passed in a queue when it receives messages from other agents. The messages are compliant with FIPA standards, discussed in 1.3.1, which are comprised of *sender*, *receiver*, *performative*, *content*, *language*, *ontology* etc. By creating *ACLMessage* objects, and calling *send()* method, messages can be sent to other agents easily. The receiving agents can receive the message by calling *receive()* method. A message also can be blocked until certain specifications are met.

3.4 Conclusion

In this chapter, the software packages that are used in this work were described. JADE and Matlab were used for agent modeling for fault detection and reconfiguration respectively. The power system was modeled using DEW. The interface between both the packages was achieved by writing the power flow results in a text file and reading it from the agent model.

Chapter 4

THE MATHEMATICAL MODEL FOR MULTI-AGENT SYSTEM

Any agent application has to be first modeled mathematically, which can be then translated to software packages for implementation. Thereby, this chapter presents a new algorithm for fault detection and reconfiguration, based on graph theory.

4.1 Mathematical Model

4.1.1 Graph Theory

Graph Theory is a study of graphs and mathematical structures that can represent any network applications. Graph is a context, refers to a collection of vertices or nodes and a collection of edges or arcs that connect pairs of vertices [60]. Many practical applications, which can be represented as networks, can be modeled based on graph

theory. Therefore, power system, which is a very complicated network, is modeled as a graph in this work.

4.1.2 Graph Theory representation of Power System

The power system network is modeled as a graph G with a single root, as shown in Figure 4-1. A spanning tree, T of G represents a radial power distribution feeder. Each node of the graph G represents either a power source, a node or a load and each edge of T represent a physical connection of different nodes through distribution lines. The edges in G , but not in T , represent switches, which are normally in open state.

Notations:

- G - graph, represents the power system network
- T - spanning tree, represents a radial feeder of the power system
- N - set of nodes in the network that is modeled as a graph G , represents source, buses, switches or loads
- E - set of directed edges, usually called arcs of G , represents the distribution lines
- s - power source, represents a substation or a DG
- $S(G)$ - set of edges in G , but not in T , represents the set of switches in the system

An agent is placed at every node and every switch.

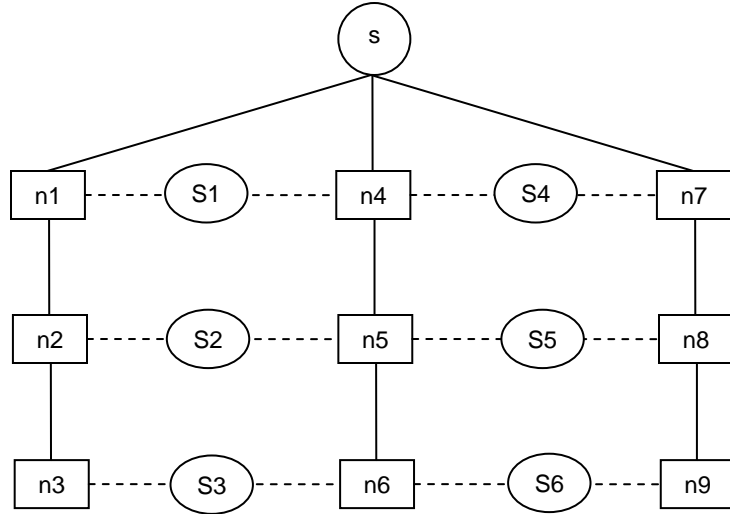


Figure 4-1: Diagram of the Notations of a Graph.

In the above graph, G , each branch $s \rightarrow n_1 \rightarrow n_2 \rightarrow n_3$, $s \rightarrow n_4 \rightarrow n_5 \rightarrow n_6$, $s \rightarrow n_7 \rightarrow n_8 \rightarrow n_9$ is a tree. The edges which are in the graph, but not in the trees are the switches, $S_1 - S_6$.

Orientation of T :

The tree, T is oriented in such a way that every node other than the source node (root) has in-degree² 1. That is, every node n in T can be reached by a directed path in T from the source s to n .

Presetting:

For each edge $e \in E(T)$, let

$$S(e) = \{X : X \subseteq S, \text{ such that } (T - e) \cup X \text{ is a connected graph}\}$$

² In-degree of a vertex in a graph is defined as the number of edges coming into the vertex.

4.2 Fault Detection Algorithm

The first step towards the reconfiguration is to identify the fault location precisely. The fault detection algorithm constantly monitors the power flowing into each load and follows the algorithm described below to identify the fault location.

Let us assume that the fault location detected by the node agents is at n_f . Suppose a node agent n identifies that the power flowing into its node is zero, then on the unique directed path from s to n , say $s \rightarrow n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \dots \rightarrow n_k$, where $n = n_k$, n_k will request n_{k-1} , whether it has power. Subsequently, n_{k-1} will request n_{k-2} for its power. When for some i , n_i does not have power, but n_{i-1} has power, the fault edge can be identified as, $e_f = (n_{i-1}, n_i)$.

Once the fault location is identified, the edge e_f is isolated and agents that are controlling each switch $X \in S(e_f)$ will communicate and coordinate with each other to decide on the particular switch or switches that have to be switched on, so that $(T - e_f) \cup X$ will continue to be a fully supplied network.

The mixed programming algorithm for reconfiguration through selection of proper switching is given in the next section.

4.3 Reconfiguration & Restoration Algorithm

It is vital to restore the power supply promptly by re-routing the power flow through a target configuration, when the power supply is interrupted by a fault. The problem of obtaining a target system is referred to as power system restoration [61].

The fault location, e_f , has been found from the previous algorithm and a collection of possible switches, $S(e_f)$ have been provided, as stated in presetting in section 4.1. Thereby, the graph $G(e_f)$ is the graph obtained by incorporating $T - e_f$ and all the

edges as a member in the collection $S(e_f)$. This algorithm assumes that each edge in the tree can be isolated in order to reconfigure the system.

Let, P_s be the total amount of available source power of the system.

For a node i , let P_{L_i} denote the active power consumed by the load at i and,

$O_i = \{(i, j) \in E : j \in N\}$, out flow from node i and,

$I_i = \{(j, i) \in E : j \in N\}$, in flow to node i .

For an oriented edge, e , let P_e denote the amount of power flowing through e , and let Y_e denote the state indicator variable:

$$Y_e = \begin{cases} 1 & \text{if } e \text{ is closed} \\ 0 & \text{if } e \text{ is open} \end{cases}$$

It is prudent to assume that for each edge $e \in (E(T) - \{e_f\})$, $Y_e = 1$. Therefore, the only fact that has to be decided is which of the switches will be closed and which should remain open.

Objective Function:

$$\text{Maximize} \quad \sum_{i \in N} W_i P_{L_i}$$

Where, W_i is the priority of the load, given as a weightage, and P_{L_i} is the supplied load.

N is the number of all load agents.

Subject to constraints:

- $P_s \geq \sum_{i \in N} P_{L_i}$

i.e. the total load should not exceed the total source capacity

- $P_{ij} \leq P_{ij, \max}$

i.e. the power flow through any arc or directed edge, e , when it is closed, should not exceed the capacity of the edge. Where P_{ij} is the real power flowing through the line connecting nodes i and j and $P_{ij,\max}$ is the maximum allowable power through that line.

A node that is not a source node is called an intermediate node.

- At a non load intermediate node i , the amount of power flowing in should be the same as the amount flowing out.

$$\sum_{e \in I_i} P_e - \sum_{e \in O_i} P_e = 0$$

- At a load node i , the amount of power flowing in should be equal to the sum of the amount flowing out and the load at i .

$$\sum_{e \in I_i} P_e - \sum_{e \in O_i} P_e = P_{L_i}$$

- The distribution system is radial, i.e., the total number of incoming edges at any node is at most unity [62].

$$\sum_{k \in I_i} e_k \leq 1$$

A reconfigured topology is achieved by selecting from the choice of $S(e_f)$, which comply with the objective function and the constraints listed above. However, when a case arises, that the model complies with multiple switching scenarios, a switching scenario is selected randomly in this model. However, this can be related with cost function to select the optimal switching scenario for reconfiguration.

4.4 Illustration

The explanation of the algorithm described above is illustrated clearly by using the simplified CoF diagram, shown in Figure 4-2. In the figure, N_i are the nodes, L_i are the loads, S_i are the switches, s is the substation and DG is the Distributed Generator modeled as a backup source. Let us assume that the fault is in $e_f = (N_1N_2)$.

The load agents in L_1 , L_2 and L_3 will detect that there is no power flowing into their respective loads. Hence, these three load agents will communicate with their respective neighbors in the incoming tree from $N_4 \rightarrow N_3 \rightarrow N_2 \rightarrow N_1$ for their flows. When all these nodes detect that they don't have power flow, the MAS can locate the fault in N_1N_2 .

Thereafter, MAS has to find a new topology for restoring the power system, avoiding the faulty line. The switching possibilities for reconfiguration, when there is a fault in $e_f = (N_1N_2)$ are:

$$S(e_f) = \{S_1, S_2, S_3, \{S_1, S_5\}, \{S_2, S_5\}, \{S_3, S_5\}, \{S_1, S_4, S_6\}, \{S_2, S_4, S_6\}, \{S_3, S_4, S_6\}\}$$

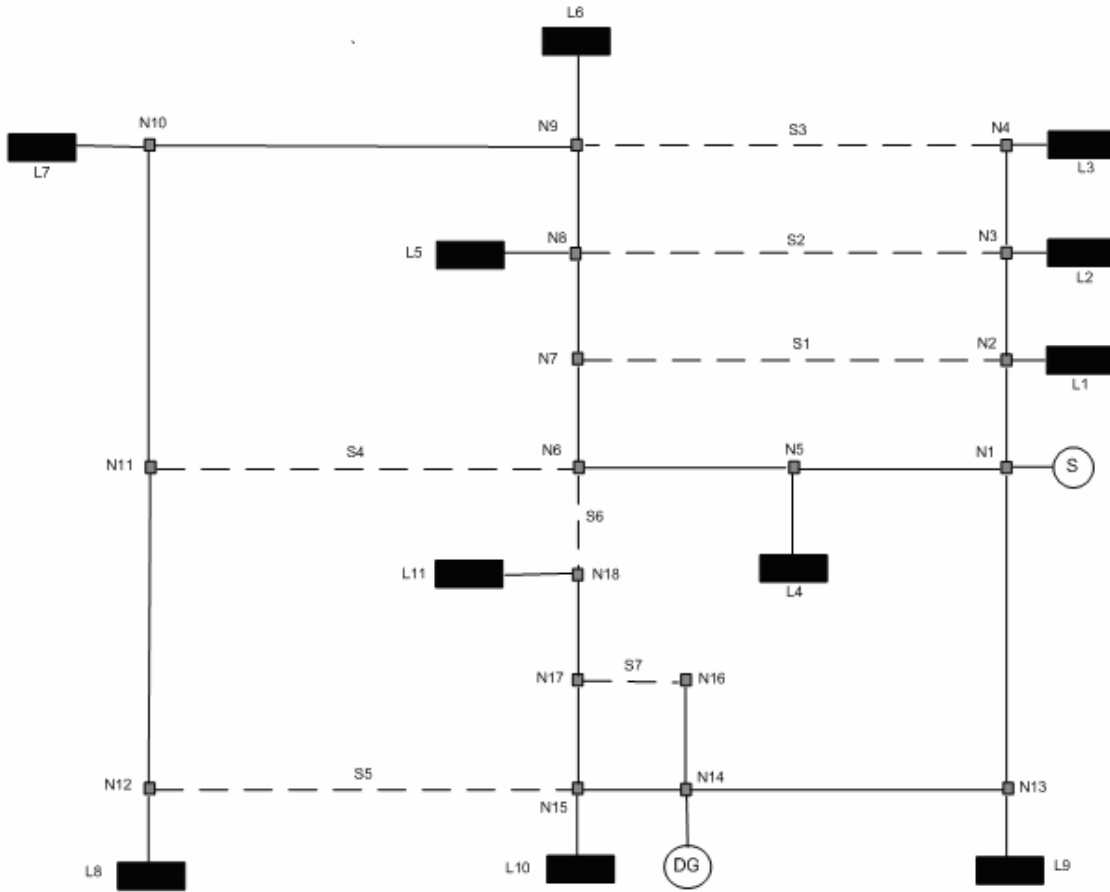


Figure 4-2: Graphical Representation of Simplified CoF.

Then the agents in these switches will communicate and negotiate with each other to find the best switching possibility, which meets the objective function and the constraints. Suppose, the best solution is to close S_1 and S_5 , the decision vector will be $\{ 1, 0, 0, 0, 1, 0, 0 \}$.

4.5 Conclusion

A mathematical model was developed based on graph theory and applied for fault detection and reconfiguration of the distribution circuit concerned. The chapter concluded with a simple illustration of a fault scenario.

Chapter 5

SIMULATION AND RESULTS

The new algorithms for fault detection and reconfiguration have been tested for different scenarios. The fault detection was simulated in JADE and reconfiguration was tested in Matlab. Both the results were combined with DEW for power system simulation and the results were tested for voltage compliance.

A general model for the application of MAS in the power system engineering is shown in Figure 5-1. The proposed MAS architecture is illustrated in Figure 5-2. It consists of Load Agents (LAGs), one for each load, and Switch Agents (SAGs), corresponding to each switch or disconnector in the system. Any LAG in the circuit can be associated with one or more SAGs. These agents coordinate with each other in order to detect the faults in the system and re-route the power flow to better serve the customers.

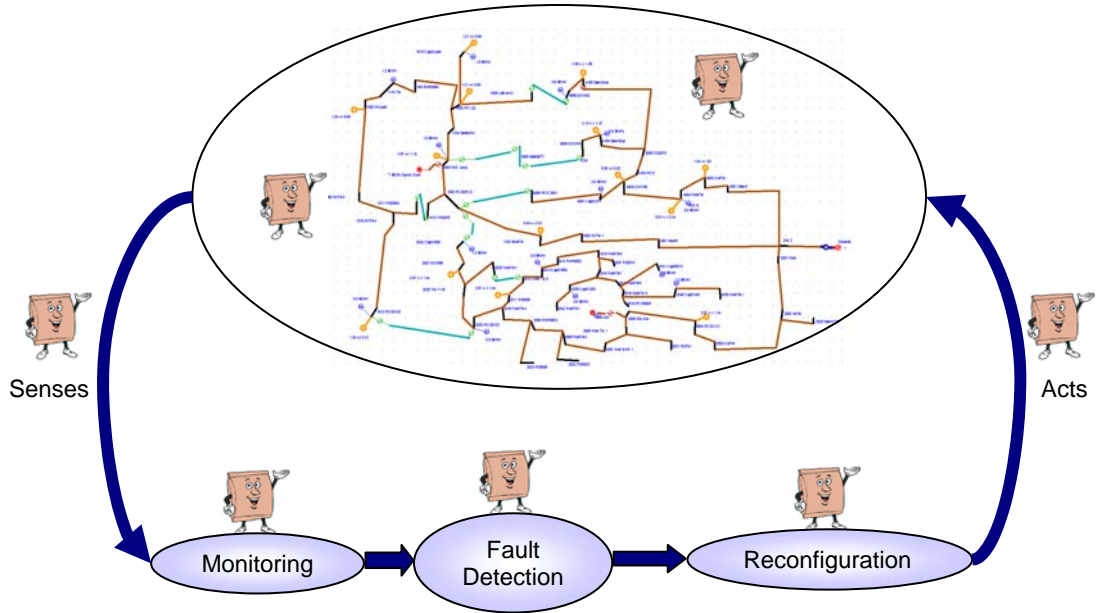


Figure 5-1: Generic model of the MAS application in Power Distribution System.

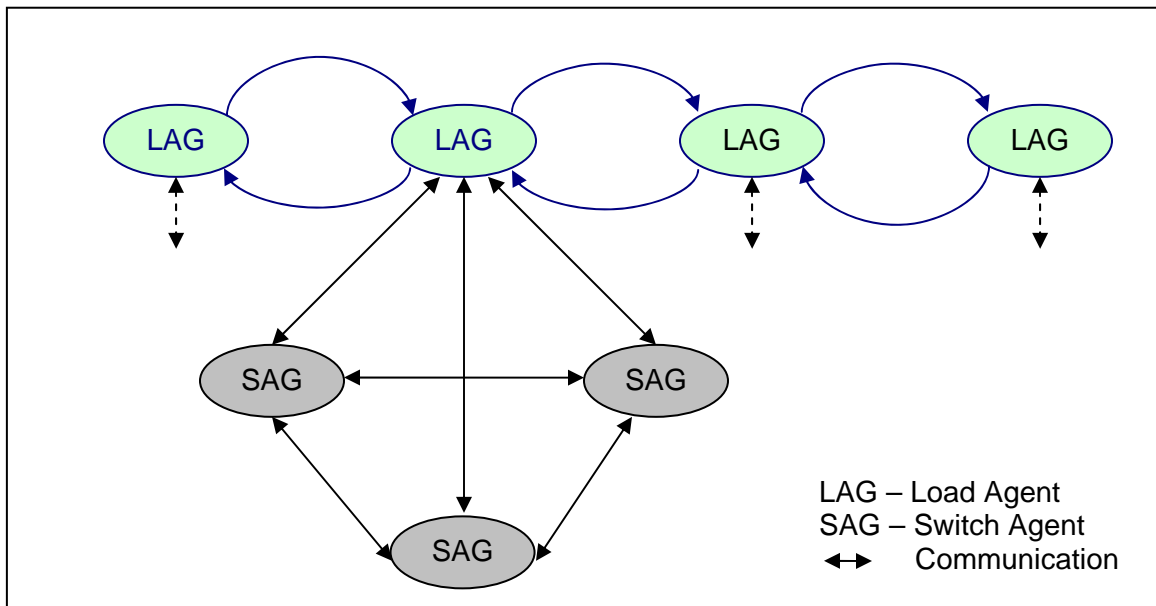


Figure 5-2: Architecture of the Proposed MAS.

5.1 Circuit of the Future

Both the simulations were carried out for a proto-type power distribution system, the Circuit of the Future (CoF), developed by Southern California Edison (SCE). The CoF modeled in DEW is shown in Figure 5-3.

The CoF has a single substation, with three main feeders, which are also connected for flexible re-routing of power flow, through 7 switches/disconnectors which are normally open. The circuit has 14 loads demanding total real power load of 24 MW and reactive power load of 12.96 MVar. It has 14 Capacitor banks for providing Var support and two DGs, one providing real power and the other reactive power. Since the capacitor banks can cater for the entire reactive power loads and losses in the system, only the active power flow is considered in the simulations.

For simplicity for simulation purposes, the original circuit is slightly modified by lumping certain loads without affecting all of its 7 switching locations and the original system topology.

The modified CoF will have 11 loads, 7 switches, 18 nodes, a Substation and a Distributed Generating Source, which provides real power. It is assumed that the power factor of the system is 0.85, which is used in calculating the real power capacity of the distribution lines. The distribution system is considered to be a radial system in all system configurations. For simulation purposes of serving the high priority loads, when there is a power shortage, three loads have been chosen to be high priority loads (L_3 , L_8 and L_{11}) and two loads have been chosen as low priority loads (L_7 and L_{10}). The rest of the loads are given medium priorities, as given in Figure 4-2.

5.2 JADE Fault Detection Simulation

JADE Fault detection simulation can be initiated either by,

- Creating JADE agents from reading the text file, which is generated by DEW after running the load flow application, or
- Creating JADE agents from java platform, by passing the necessary parameters for each agent, as command line arguments.

Both the above methods were tested for correctness and the second method is used in the simulations in this chapter.

A flow chart for fault detection process is given below.

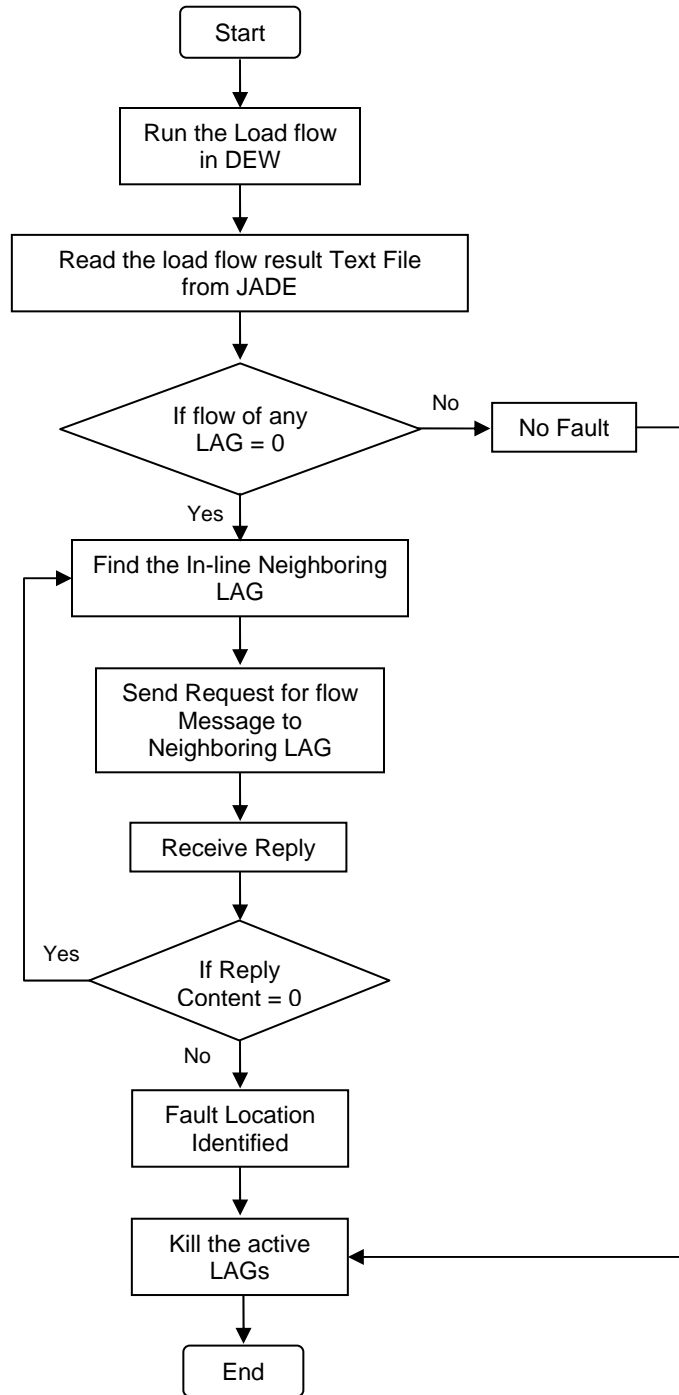


Figure 5-4: Flow chart for Fault Detection.

5.2.1 Test Case 1: For a single fault

Let us test the system for a fault in line N_1N_2 in Feeder 1, as depicted in Figure 4-2.

When the JADE MAS is run, the relevant load agents communicate with each other to find out the fault location. The message passing is clearly shown in JADE Sniffer agent GUI.

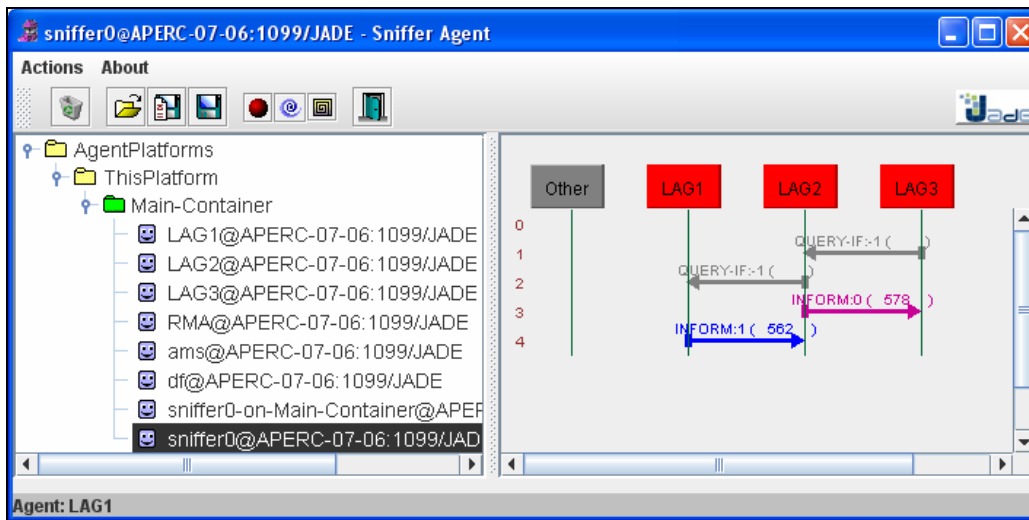


Figure 5-5: JADE Message Passing to identify the fault location for a single fault in N_1N_2 .

All the LAGs will be created in the particular feeder. But only those LAGs which identify, that they have no power flowing in, will send a *request* message to its incoming neighboring agents whether the neighboring agents have their loads supplied or not and wait for their reply. In this case, all three LAGs, do not have power flowing into their loads due to the fault at N_1N_2 . Hence the LAGs which received the *request* message reply to the sender with their flow values. Since all the LAGs have zero flows, the fault location can be identified accordingly.

In case, the fault is in N_2N_3 , LAG3 and LAG2 will not have their loads supplied. Hence, they initiate the communication to their neighbors LAG2 and LAG1, respectively. In this case, LAG2 will respond with 'zero', whereas LAG1 will send its actual power flow, which is non zero floating value. Thereby, the fault location can be identified as the link between LAG1 and LAG2; that is the line N_2N_3 .

5.2.2 Test Case 2: For Multiple Faults

For testing a multiple fault scenario, let us assume that there are 3 faults in all three feeders being at N_2N_3 , N_1N_5 , and at $N_{13}N_{15}$. This situation has been run in JADE fault detection program and the message passing between the LAGs have been observed to be as shown in the following figure.

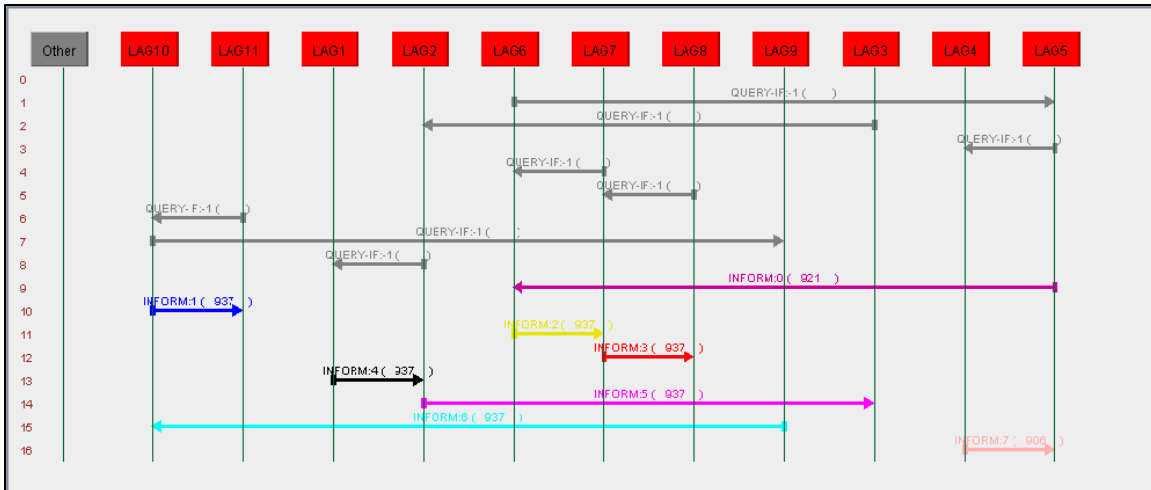


Figure 5-6: Sniffer Agent GUI for Multiple Fault Scenarios.

The finding of the fault locations is given in the java console as given below.

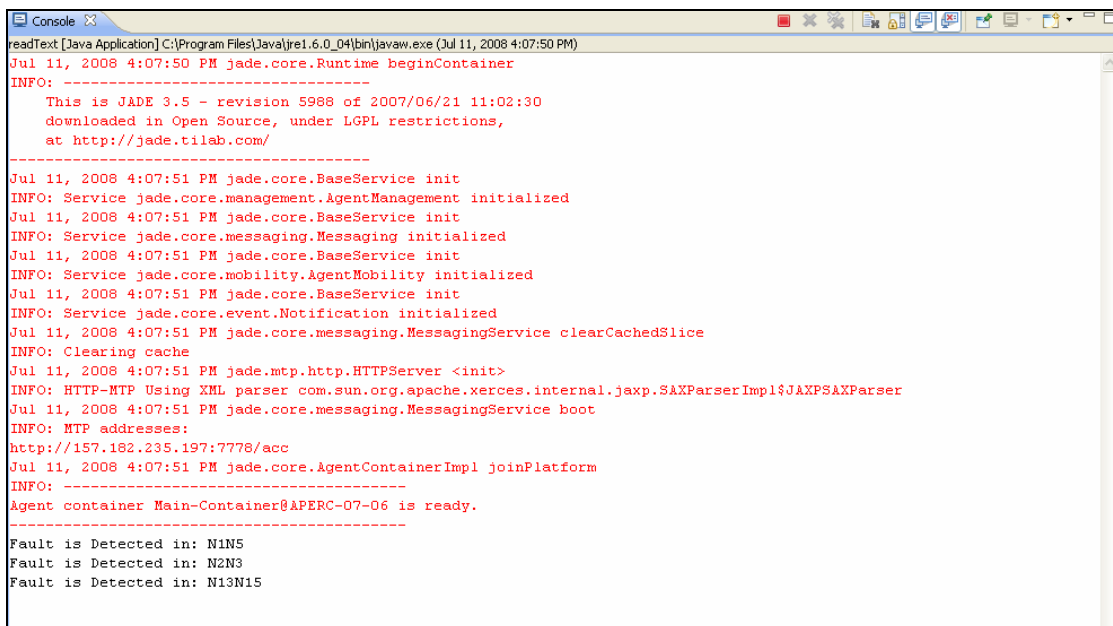


Figure 5-7: Output screen of the solution for Test Case 2.

5.3 Matlab Reconfiguration Simulation

Reconfiguration of the CoF has been implemented in Matlab using graph theory, as described in chapter 4. As described in the flow chart, given in Figure 5-10, each fault location will be identified by the relevant LAG, and will have corresponding combination of SAGs for reconfiguration. Each proposal is reconfigured using the Energy Management System used in the thesis by Pinak Tulpule [63], to find the best solution for CoF. All the possible fault cases have been tested in the model. A few different scenarios have been presented in the simulations.

Reconfiguration is carried out based on maximum flow algorithm. The maximum flow principle is a branch of graph theory and combinatorial optimization. It seeks a feasible solution that sends the maximum possible flow from a source to a sink node. In the reconfiguration of energy management system, maximum flow algorithm represents the power system as graph orientations [63].

5.3.1 Illustration of Reconfiguration with IntelliTEAM® II

An example for reconfiguration is illustrated below for the IntelliTEAM II circuit. This circuit, introduced by S&C Electric Company has three sources and three loads. It can be observed, that the circuit under normal operations, as given in Figure 5-8, is fed by source 2, which is considered as the primary source. The other two alternate sources start to supply only when the necessity arises. Load at Sink 7 is given the highest priority.

For a case where there is a fault in the primary supply line, the reconfiguration is run to supply maximum load as possible by adhering to the capacity limits of the lines. Thereby, the reconfigured circuit is given as shown in Figure 5-9. It can be observed that due to the fault, all three loads are now being supplied from both the alternate sources, since any single source other than the primary source, is not adequate to cater for the total loads.

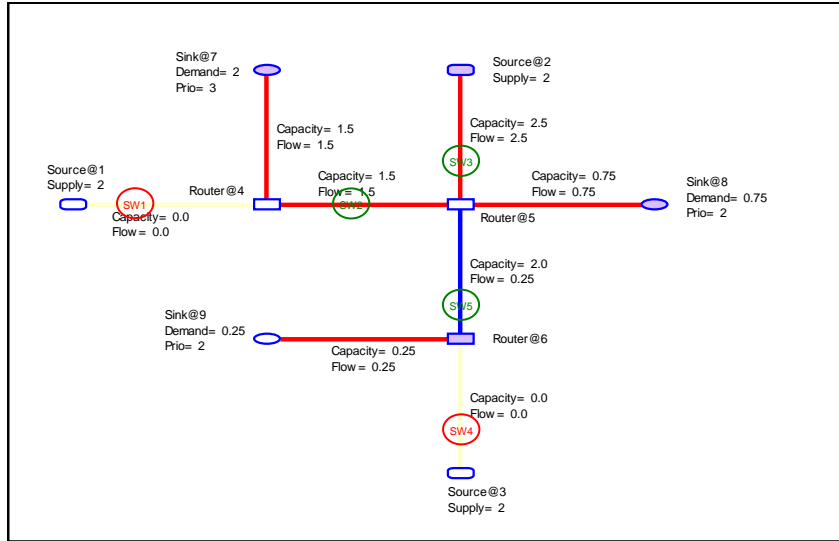


Figure 5-8: IntelliTEAM® II under normal operations.

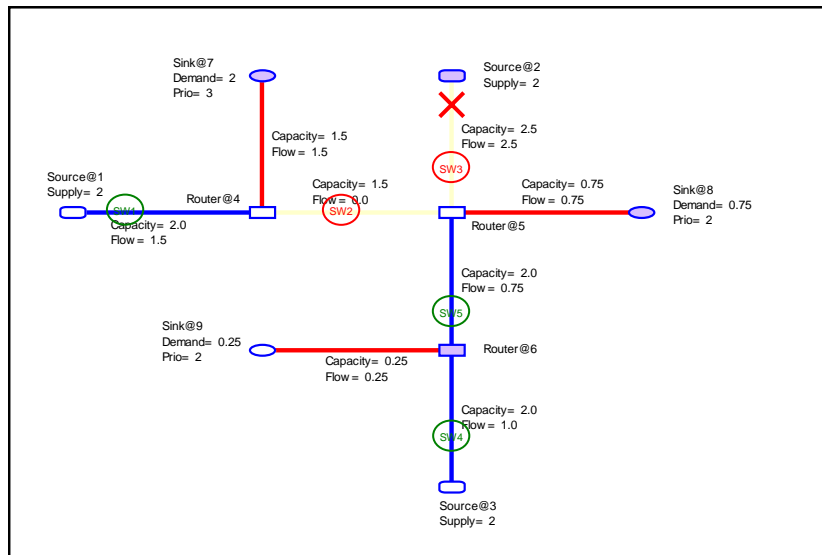


Figure 5-9: Reconfigured IntelliTEAM® II after the fault in the primary supply line.

The above fault scenario is simulated using Energy management system [63], developed at WVU. In order to select the target configuration of the CoF, the model has been enhanced in order to choose the proper switching positions from the available multiple supply paths. Hence, a new reconfiguration model is proposed accordingly, in this work, as described previously and as shown in Figure 5-10.

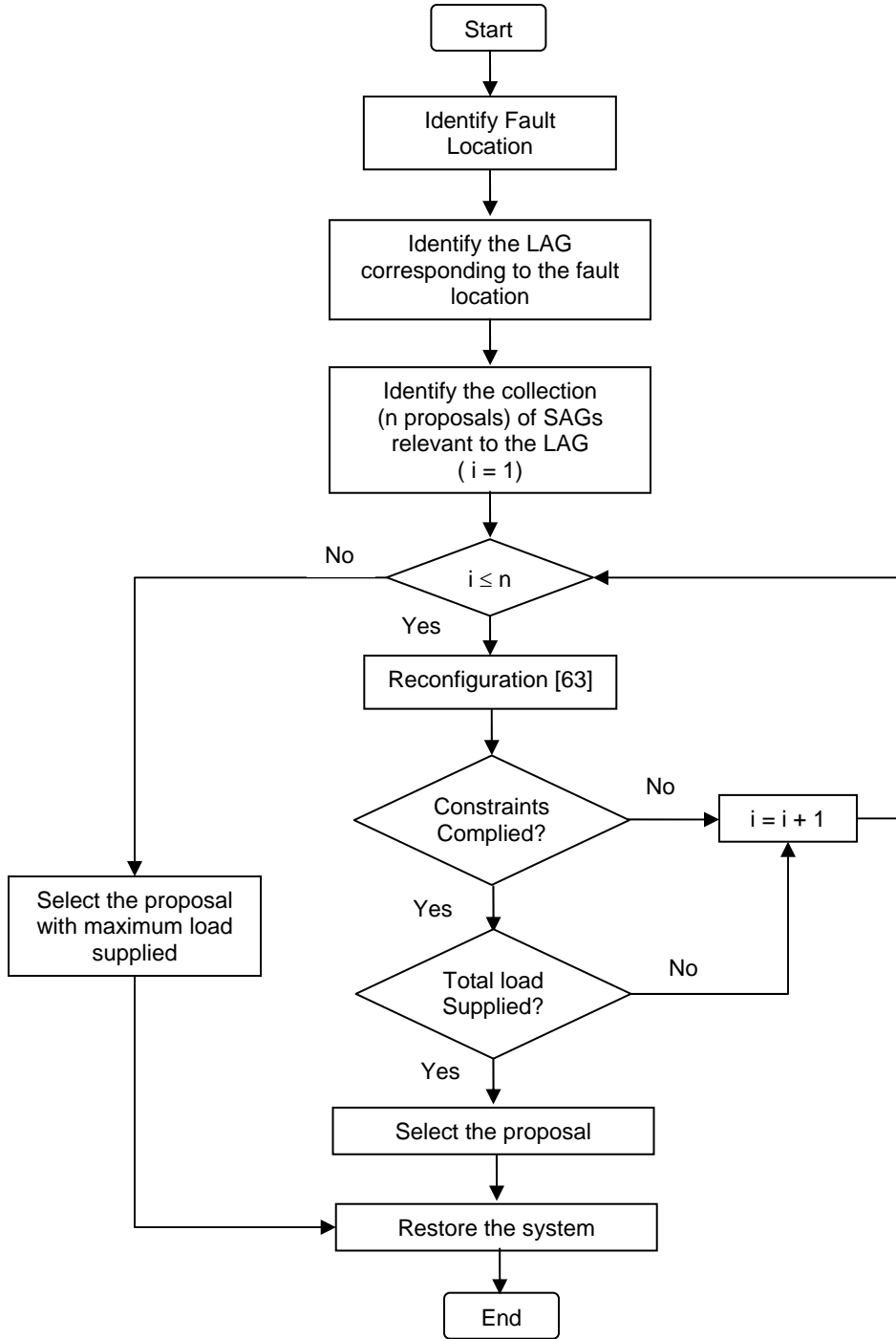


Figure 5-10: Flow Chart for Reconfiguration.

A GUI is developed for easy user interaction for carrying out simulations. The GUI is given in the figure below.

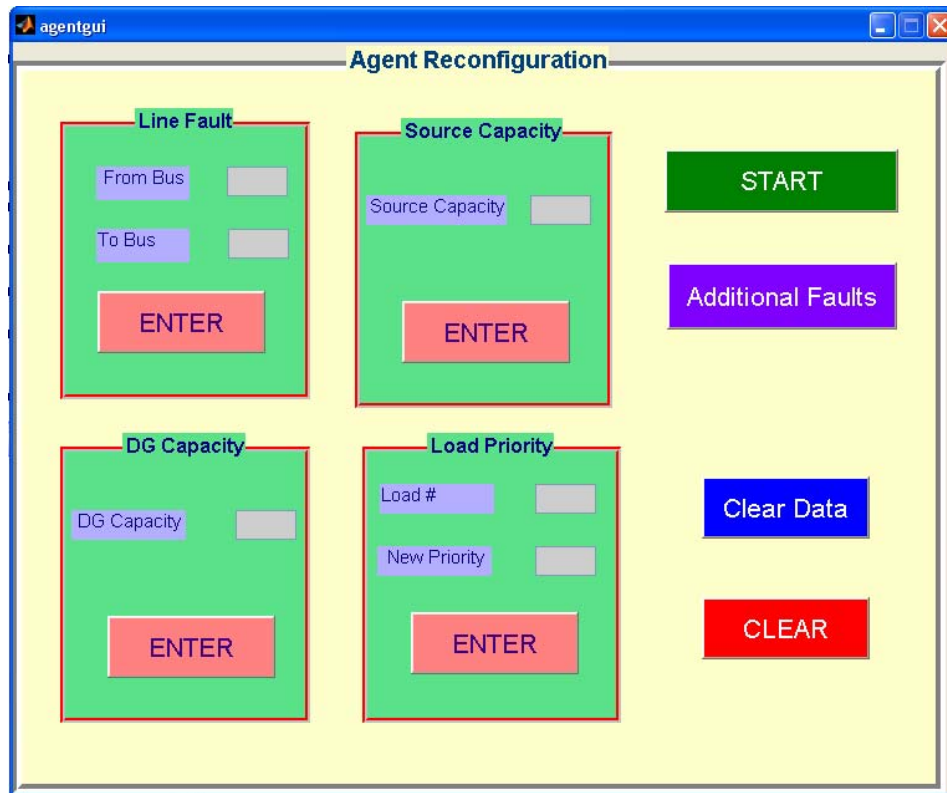


Figure 5-11: GUI for user interaction for Reconfiguration.

It can be seen that four scenarios can be developed. A line fault can be created by specifying the 'from node' and the 'to node' agent names. The source capacity can be decreased to test the shortage of supply from the substation, by entering the new capacity in kW. It is also possible to change the capacity of the DG and the priorities of the loads for testing purposes.

Multiple fault scenarios can be simulated by entering the specific data and pressing on 'Additional Faults' button. 'Clear Data' button clears the data entered in the GUI and 'CLEAR' button clears all the data in the memory to start a fresh scenario.

The normal operation of the circuit is given below before any faults.

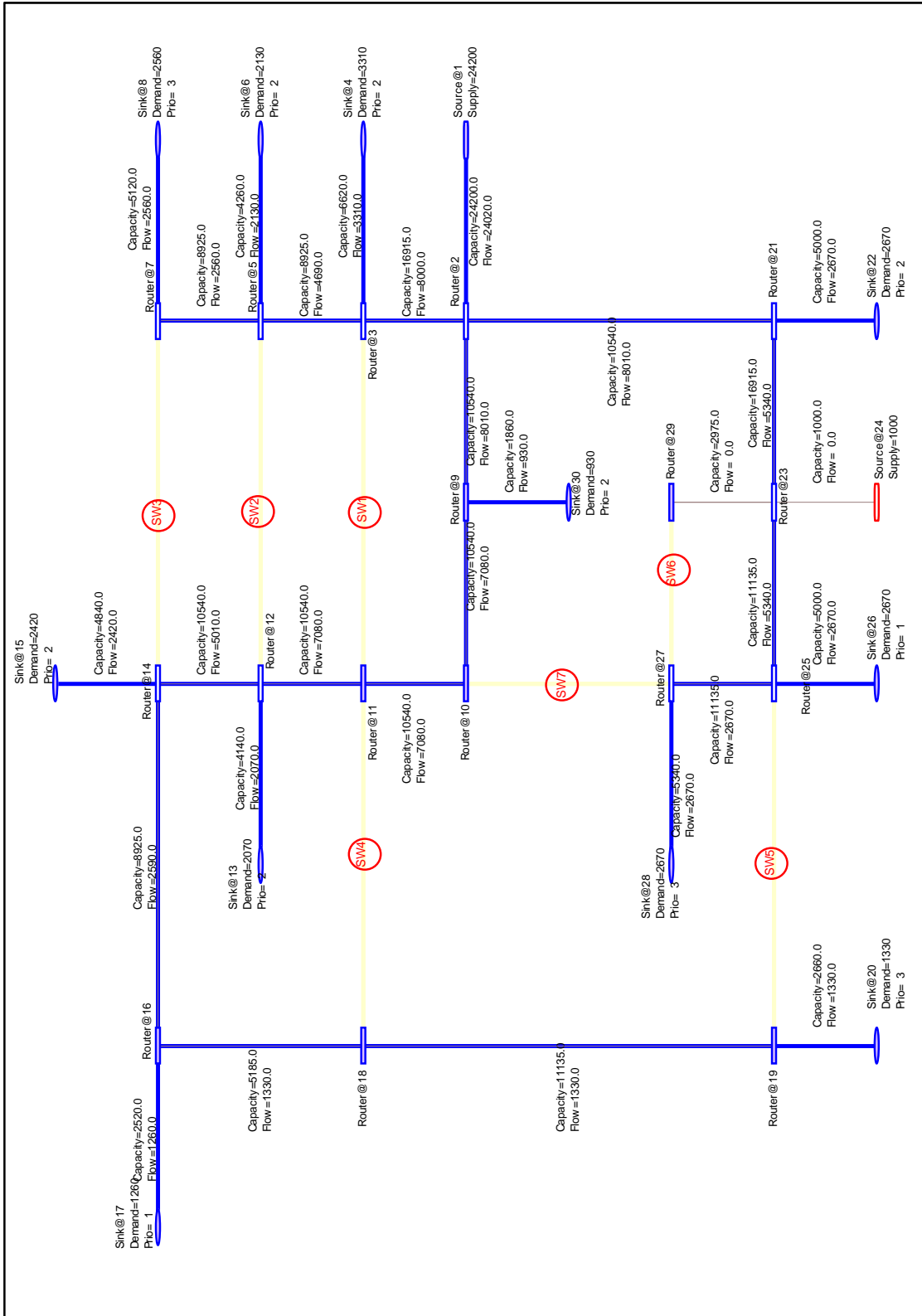


Figure 5-12: Graphical representation of the CoF under normal operations.

5.3.2 Test Case 3: For a Line Fault

Assume that a fault is identified between agent numbers 2 and 3 in Figure 5-12. The algorithm is run to reconfigure the system in order to achieve the objective function and satisfy all the constraints, explained in the previous sections. Figure 5-13 shows the reconfigured circuit.

It can be seen that, from all the switching possibilities, switch 1 and 5 are operated in order to supply the loads in the faulty feeder. However, since, this fault is on one of the three main feeders, the total load of 24 MW, cannot be supplied due to capacity constraints of the other two main feeders. It also can be noticed, that within the available capacity, all the high and medium priority loads are supplied and only the least prioritized loads are not supplied fully.

Out of all the possible line faults that can occur in the CoF, this particular fault, is the least desirable fault, where the total load cannot be supplied without overloading the distribution lines. In all other line fault cases, combination of switches can be found for system reconfiguration, without curtailing the power supply to any loads.

5.3.3 Test Case 4: Shortage of Source Capacity

In the event that the source has a shortage of capacity, the new topology of the system is found in such a way to ensure that with the available source capacity, first the high priority loads are supplied. In case, there is still remaining source power available, the lower priority loads are supplied in the priority order.

In this test case, it is assumed, that the source capacity is reduced to 3 MW, from 24 MW. From the reconfigured network, shown in Figure 5-14, it can be observed that 3 MW is not even adequate to supply the high priority loads. Hence, to supply the remaining high priority loads, the DG is switched on. However, since the source and DG have a total source of 4 MW, but the high priority loads add up to 6.56 MW, even the total prioritized loads cannot be supplied fully with both the substation and the DG sources.

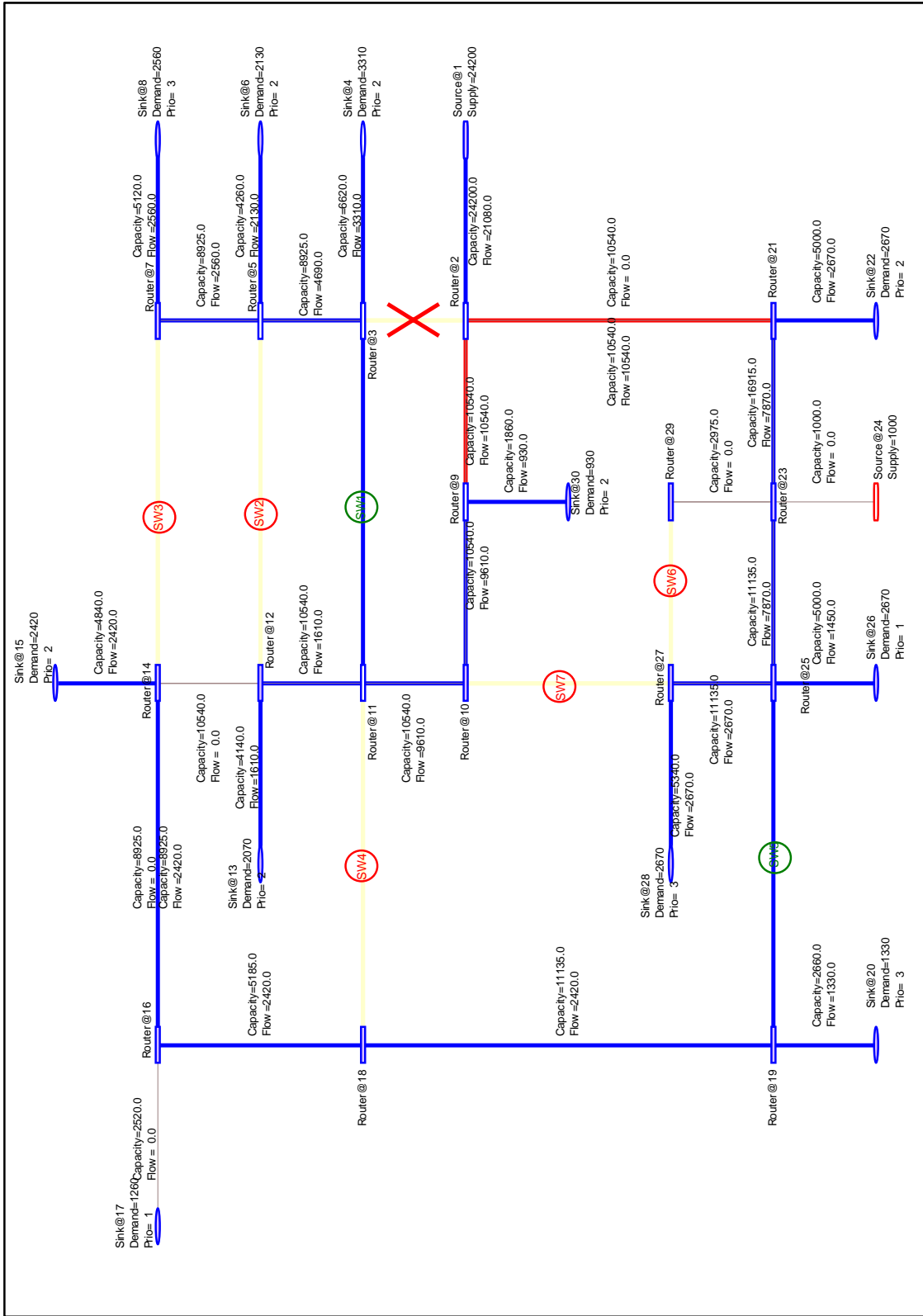


Figure 5-13: Reconfigured System after a line fault.

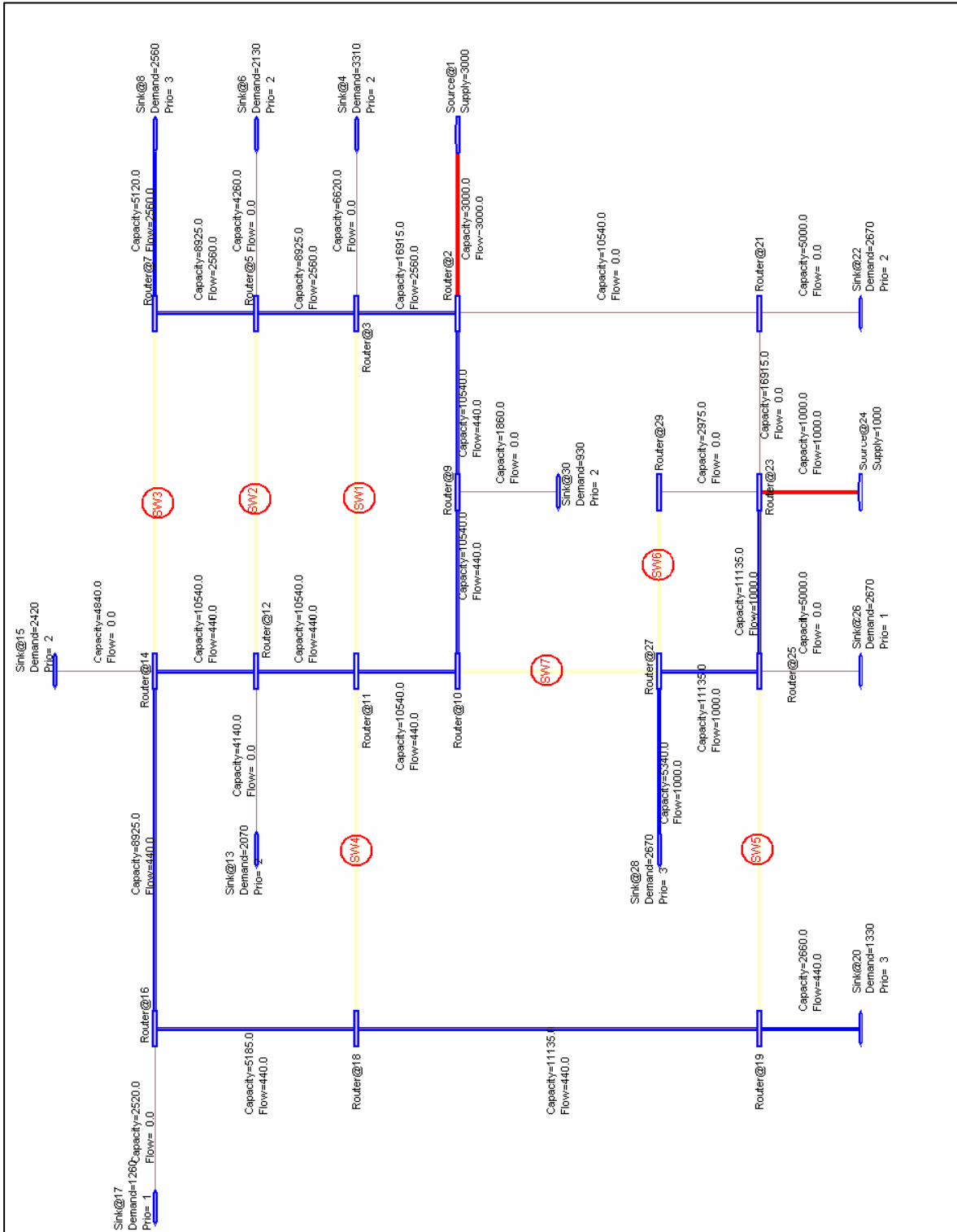


Figure 5-14: Reconfigured Circuit during a shortage of source capacity.

5.4 Algorithm Execution Duration

Both the fault detection and reconfiguration algorithms are tested for the duration of operation. Time durations obtained from the testing of several executions are plotted as shown below. Figure 5-15 shows the execution time taken for 100 samples of fault detection processes. It can be seen that the average time taken for execution of fault detection is 22.4 ms.

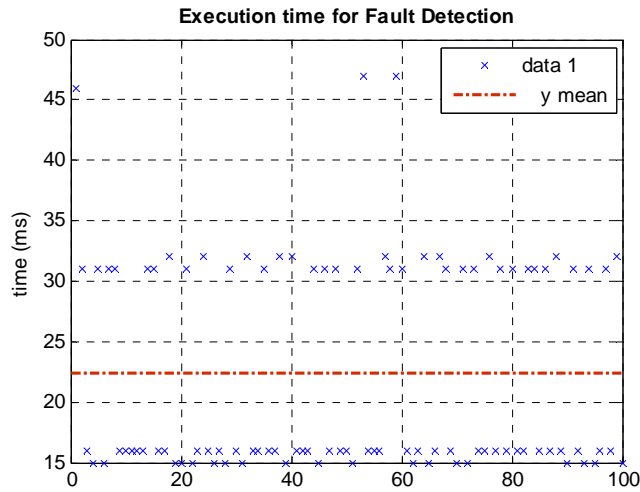


Figure 5-15: Plot of Execution Time for Fault Detection Algorithm.

Figure 5-16 shows the execution time taken for reconfiguration for 50 sample executions. The average time taken for the reconfiguration process is 14.5 seconds.

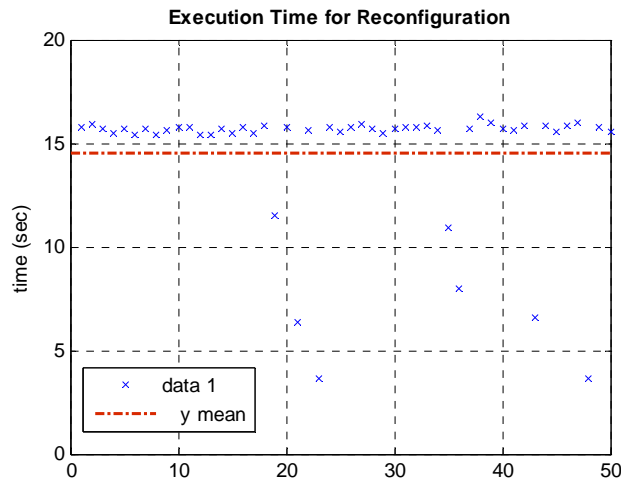


Figure 5-16: Plot of Execution Time for Reconfiguration Algorithm.

It has to be noted that the average execution times calculated are for the proto type circuit simulated in this work. But in reality, the execution time will be higher than the software simulation durations given in this section. This is due to the time taken for interfacing the software with the hardware and also due to the actual time delay in communication links that are used for message transportation.

5.5 Testing the Voltage Profile

It is important to test the voltage profile of the reconfigured system to check whether the customer voltages are maintained within the tolerance limit. Hence the original system and the Test Case 3 were simulated in DEW to analyze the voltage profile in Figure 5-17 and Figure 5-18 respectively.

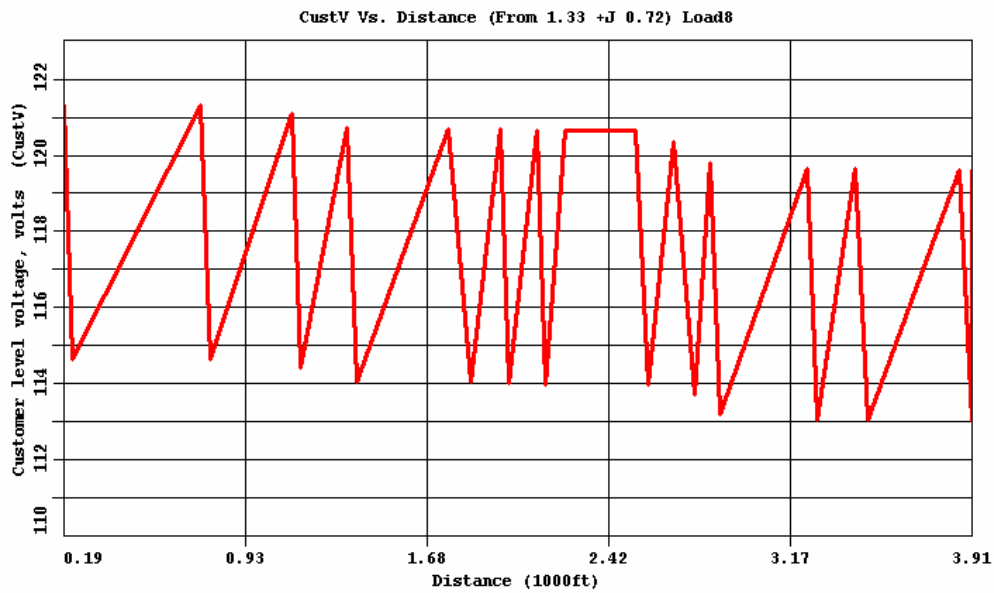


Figure 5-17: Voltage profile of the CoF under normal operations.

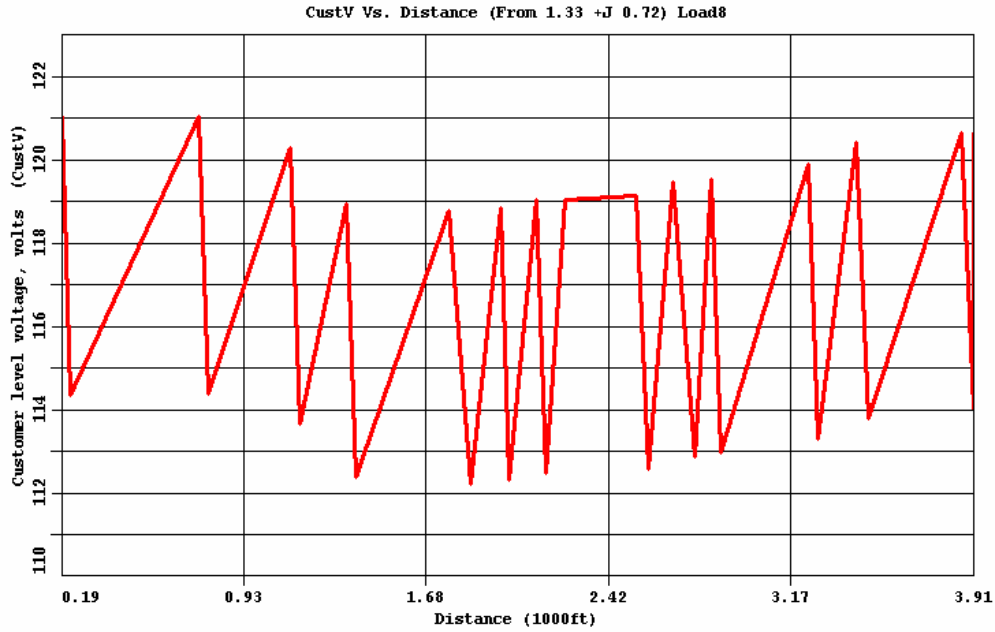


Figure 5-18: Voltage profile of the CoF under the faulty situation in Test Case 3.

From the above graphs, it is shown that by reconfiguring the system, the system voltages are maintained within its limits of $\pm 10\%$ tolerance.

The reason for the voltage peaks and dips observed in the graphs above are due to the capacitance of the distribution cables in the system.

5.6 Conclusion

Simulations carried out in this section analyze a few test fault cases. In addition, the algorithm has provision to change the capacity of the DG and also to change the priority of the loads, if required.

All the possible cases were tested in the software implementation and the results were as expected.

Chapter 6

CONCLUSION AND FUTURE WORK

Application of MAS in power system operations are being widely researched at present, due to the increasing complexity of the power system. Power network are becoming larger and the present control mechanisms are not efficient enough to cater the increasing intricacy. A better mechanism to solve this issue is by application of MAS. A MAS is intelligent, decentralized and provides speedy solution with less human intervention for today's need.

6.1 Conclusion

MAS models for fault detection and reconfiguration applications for a proto-type circuit are presented in this work. The circuit concerned is the Circuit of the Future, a power distribution system that has three main feeders, several loads and switches.

The switches are usually open and are operated only during a fault in any other lines, to maintain the power supply to all the loads. To cater for these switching operations, a new mathematical model is developed based on graph theory. The decision making is carried out by Load Agents and Switch Agents, corresponding to the fault location and fault type. These agents communicate, coordinate and negotiate with each other in a decentralized manner, with the available local information to each of them. The ultimate decision is based on the local coordination of the agents in the system. In reality, each component of the power system has to have an agent implemented in order to perform their individual tasks to attain the global objective and to abide by the constraints of the system.

The main objective of the system is to supply as much load as possible in any fault situation, such as line fault or shortage of supply. Hence, the reconfiguration model has to be designed to re-route the power flow through the possible switching operations.

The fault detection model that is developed in JADE can find the exact fault location by communication and coordination of LAGs. The reconfiguration model, developed in Matlab finds out the best switching operation in order to achieve the overall objective of the system.

All possible fault scenarios were tested in both the models. The results obtained are promising and it shows a very good start in the direction of MAS application in power distribution system.

The simulations reveal that the critical line fault that can occur is the fault in line N_1N_2 . If the fault location is in the beginning of the first feeder, 12.24% of the total load cannot be supplied. This harms the system reliability and can be avoided if the distribution line capacity can be increased.

The execution times were tested for both the fault detection and reconfiguration algorithms and the average duration is found to be 22.4 ms and 14.5 s respectively. The voltage profiles were also observed for all the fault cases and are observed to be within its tolerance limits.

6.2 Future Work

The research work in this area is in the initial stages and hence, there is a lot of possible extension in the work which can be applied to run the entire power system with MAS.

The very next steps of work that are intended to expand are:

- Interfacing the power system application and JADE agent model for real time operation. This can be achieved by finding an interface between DEW, which is developed in C++ and JADE that runs in java platform or by developing a power system model entirely in java platform, which can be easily operated by JADE agent model.
- Major assumption in this work is that the system is always maintained in radial topology. In reality, in some situations, distribution systems in big cities or system with DGs, this assumption is not true. Hence, a new model has to be developed to accommodate for loop or network system configuration.
- An agent model for Voltage Regulation has to be designed. This is a very interesting step towards the research area of MAS applications in power system. In distribution system, the voltage must be maintained within the tolerance limits in order to maintain power quality and to avoid system collapse due to voltage sags. Hence, this future work will be really rewarding.
- Cost can be associated with fault cases, so that when several viable proposals are received by an initiator agent while performing reconfiguration, it can select a proposal with minimum cost.
- Agent Learning can be applied for agents to act based on past experience. When a fault occurs, and the agents perform a certain task to achieve their objective. The MAS can learn from this action, and have this in its memory. In case, the same situation arises in the future, the agents can recollect the situation from their memory, and without performing the assigned tasks, it can directly provide

the solution. This can increase the efficiency of the system by reducing the time and operations of agents.

- Finally, all the agent applications designed, such as fault detection, reconfiguration, voltage control, reactive power management, incorporation of DGs, protection coordination, learning etc. shall be incorporated together to perform the entire tasks required to upgrade the power system to be more reliable.

REFERENCES

- [1] Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Power_grid#History
- [2] Wikipedia, http://en.wikipedia.org/wiki/Northeast_Blackout_of_2003
- [3] The free online Dictionary of Computing,
<http://foldoc.doc.ic.ac.uk/foldoc/Dictionary.gz>
- [4] M. Wooldridge, N. R. Jennings, "Intelligent agents: Theory and Practice," The Knowledge Engineering Review, Vol. 10, Issue 2, pp 115 – 152, 1995
- [5] G. Weiss, "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence," The MIT Press, 2000
- [6] M. Wooldridge, "An Introduction to Multiagent Systems," John Wiley and Sons Ltd., 2002
- [7] The Foundation for Physical Intelligent Agents, <http://www.fipa.org>
- [8] Wikipedia,
http://en.wikipedia.org/wiki/Foundation_for_Intelligent_Physical_Agents
- [9] D. A. Cartes and S. K. Srivastava, "Agent Applications and their future in Power Industry," IEEE Power Engineering Society General Meeting, pp 1 – 6, June 2007
- [10] Z. Zhang, J. D. McCalley, V. Vishwanathan and V. Honavar, "Multi-agent System Solutions for Distributed Computing, Communications and Data Integration

References

- Needs in Power Industry,” IEEE Power Engineering Society General Meeting, Vol. 1, pp 45 – 49, June 2004
- [11] L. Liu, K. P. Logan, D. A. Cartes and S. K. Srivastava, “Fault Detection, Diagnostics and Prognostics: Software Agent Solutions,” IEEE Transactions on Vehicular Technology, Vol. 56, No 4, pp 1613 – 1622, July 2007
- [12] R. Isermann, “Process fault detection based on modeling and estimation methods – A survey,” Automatica, Vol. 20, No. 4, pp 387 – 404, July 1984
- [13] R. Isermann, “Model based fault detection and diagnosis – Status and Applications,” Annual Rev. Control, Vol. 29, pp 71 – 85, 2005
- [14] E. Chow and A. Willsky, “Analytical Redundancy and the design of robust failure detection systems,” IEEE Transactions on Automatic Control, Vol. 29, No. 7, pp 603 – 614, July 1984
- [15] R. J. Patton and J. Chen, “Advances in Fault Diagnosis using Analytical Redundancy,” IEE Proceedings of Integrated Operations Management Control, pp 6/1 – 6/12, 1993
- [16] X. Ding and P. M. Frank, “Fault Detection via Factorization Approach,” Systems Control Lett., Vol. 14, No. 5, pp 431 – 436, June 1990
- [17] C. Angeli, “Online fault Detection Techniques for Technical Systems: A Survey,” International Journal in Computer Science Applications, Vol. 1, No. 1, pp 12 – 30, 2004
- [18] K. Huang, S. K. Srivastava, D. A. Cartes and L. Liu, “Agent Solutions for Navy Shipboard Power Systems,” IEEE International Conference on Systems Engineering, pp 1 – 6, April 2007

References

- [19] J. M. Solanki, N. N. Schulz and W. Gao, "Reconfiguration for Restoration of Power Systems using Multi-Agent System," Proceedings of the 37th Annual North American Power Symposium, pp 390 – 395, October 2005.
- [20] J. M. Solanki, S. Khushalani and N. N. Schulz, "A Multi-Agent Solution to Distribution Systems Restoration," IEEE Transactions on Power Systems, Vol. 22, No. 3, pp 1026 – 1034, August 2007
- [21] J. G. Gomez-Gualdron and M. Velez-Reyes, "Simulating a Multi-Agent based Self-Reconfigurable Electric Power Distribution System," IEEE COMPEL Workshop, Rensselaer Polytechnic Institute, NY, July 2006
- [22] J. G. Gomez-Gualdron, M. Velez-Reyes and L. J. Collazo, "Self-Reconfigurable Electric Power Distribution System using Multi-Agent Systems," Electric Ship Technologies Symposium, pp 180 – 187, May 2007
- [23] K. Huang, S. Srivastava and D. Cartes, "Decentralized Reconfiguration for Power Systems using Multi Agent System," 1st Annual IEEE Systems Conference, pp 1 – 6, April 2007
- [24] T. Nagata, H. Sasaki and R. Yokoyama, "Power System Restoration by Joint Usage of Expert System and Mathematical Programming Approach," IEEE Transactions on Power Systems, Vol. 10, No. 3, pp 1473 – 1479, August 1995
- [25] T. Nagata, N. Iba, Y. Tao and H. Sasaki, "A New Power System Restoration Framework by means of Multi-agent Approach," IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 3, pp. 1695 – 1700, October 2002
- [26] T. Nagata and H. Sasaki, "A Multi-agent Approach to Power System Restoration," IEEE Transactions on Power System, Vol. 17, No. 2, pp 457 – 462, May 2002

References

- [27] T. Nagata, Y. Tao and H. Fujita, "An Autonomous Agent for Power System Restoration," IEEE Power Engineering Society General Meeting, Vol. 1, pp 1069 – 1074, June 2004
- [28] T. Nagata, Y. Tahara and H. Fujita, "An Agent Approach to Bulk Power System Restoration," IEEE Power Engineering Society General Meeting, Vol. 1, pp 599 – 604, June 2005
- [29] T. Nagata, Y. Tao, Y. Tahara and T. Aoyama, "Development of Bulk Power System Restoration Simulator by means of Multi-agent Approach," The 47th IEEE International Midwest Symposium on Circuits and Systems, Vol. 2, pp 337 – 340, July 2004
- [30] D. Liu, Y. Chen, G. Shen and Y. Fan, "Multi-agent based Approach for Modeling and Simulation of Bulk Power System Restoration," IEEE/PES Transmission and Distribution Conference and Exhibition, pp 1- 6, 2005
- [31] L. M. Tolbert, H. Qi and F. Z. Peng, "Scalable Multi-Agent System for Real-Time Electric Power Management," IEEE Power Engineering Society Summer Meeting, Vol. 3, pp 1676 – 1679, July 2001
- [32] T. Hiyama, D. Zuo and T. Funabashi, "Multi-agent based Control and Operation of Distribution System with Dispersed Power Sources," IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 3, pp 2129 - 2133, 2002
- [33] Y. Qudaih and T. Hiyama, "Reconfiguration of Power Distribution System using Multi Agent and Hierarchical based Load following Operation with Energy Capacitor System," International Power Engineering Conference, pp. 223 – 227, December 2007
- [34] D. V. Coury, J. S. Thorp, K. M. Hopkinson and K. P. Birman, "An Agent based Current Differential Relay for use with a Utility Intranet," IEEE Transactions on Power Delivery, Vol. 17, No. 1, pp 47 – 53, January 2002

References

- [35] J. A. Hossack, J. Menal, S. D. J. McArthur and J. R. McDonald, "A Multi Agent Architecture for Protection Engineering Diagnostic Assistance," IEEE Transactions on Power Systems, Vol. 18, No. 2, pp 639 - 647, May 2003
- [36] J. Ahang and Q. Liu, "Application of a Multi-Agent Technology in Relaying Protection," International Conference on Power System Technology, pp 1 – 6, October 2006
- [37] S. J. Park and J. T. Lim, "Modeling and Control of Agent-based power protection systems using supervisors," IEE Proceedings on Control Theory and Applications, Vol. 153, Issue 1, pp 92 – 98, January 2006
- [38] Z. Xiangjun, K. K. Li, W. L. Chan and S. Sheng, "Multi-Agents based Protection for Distributed Generation Systems," Proceedings of the IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, Vol. 1, pp 393 – 397, April 2004
- [39] H. Wan, K. K. Li and K. P. Wong, "An Multi-Agent Approach to Protection Relay Coordination with Distributed Generators in Industrial Power Distribution System," Conference Record of the Industry Applications 40th IAS Annual Meeting, Vol. 2, pp 830 – 836, October 2005
- [40] H. Wan, K. K. Li and K. P. Wong, "Multi Agent Application of Substation Protection Coordination with Distributed Generators," Proceedings of International Conference on Future Power Systems, November 2005
- [41] M. W. David and K. Sakurai, "Protecting the Electric Power Infrastructure with a Mobile Agent Monitoring and Analysis System," Proceedings of International Conference on Information and Communication Technologies: From Theory to Application, pp 497 – 498, April 2008

References

- [42] W. Hu, L. Benedicenti and R. Paranjape, "Mobile Agent Network Management System Performance Study in Frame Relay Network," Proceedings of Canadian Conference on Electrical and Computer Engineering, Vol. 3, pp 1499 – 1504, May 2002
- [43] T. Shono, K. Sekiguchi, T. Tanaka and S. Katavarna, "A Remote Supervisory System for a Power System Protection and Control Unit Applying Mobile Agent Technology," Proceedings of IEEE/PES Transmission and Distribution Conference and Exhibition, Vol. 1, pp 148 – 153, October 2002
- [44] H. F. Wang, "Multi-agent Co-ordination for the Secondary Voltage Control in Power System Contingencies," IEE Proceedings on Generation, Transmission and Distribution, Vol. 148, Issue 1, pp 61 – 66, January 2001
- [45] S. Gehao, J. Xiuceng and Z. Yi, "Optimal Coordination for Multi-agent based Secondary Voltage Control In Power System," IEEE/PES Transmission and Distribution Conference and Exhibition, pp 1 – 6, 2005
- [46] 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, and Technical Challenges," IEEE Transactions on Power Systems, Vol. 22, No. 4, pp 1743 – 1752, November 2007
- [47] 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, and Tools for Building Multi-agent Systems," IEEE Transactions on Power Systems, Vol. 22, No. 4, pp 1753 – 1759, November 2007

References

- [48] G. Morejon, S. K. Srivastava and D. A. Cartes, "Integrating Virtual Test Bed and JADE Agent Platform using CORBA for Reconfiguration of Shipboard Power System," IEEE Power Engineering Society General Meeting, June 2006
- [49] Object Management Group Website, <http://www.omg.org/>
- [50] A. Aitouche and S. Hayat, "Multi-agent Model using Colored Petri Nets for the Regulation Traffic of an Automated Highway," IEEE Proceedings on Intelligent Transportation Systems, Vol. 1, pp 37 – 42, 2003
- [51] V. Gorodetsky, O. Karsaev, V. Kupin and V. Samoilov, "Agent based Air Traffic Control in Airport Airspace," IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp 81 – 84, 2007
- [52] K. Zafar, S. B. Qazi and A. R. Baig, "Mine Detection and Route Planning in Military Warfare using Multi Agent System," IEEE Proceedings of the 30th Annual International Computer Software and Applications Conference, Vol. 2, pp 327 – 332, September 2006
- [53] L. HaiHua, Z. Miao Liang, "MAS4AMR: A Self Organized Multi Agent System designed for Auto Mobile Robot," IEEE Third International Conference on Natural Computation, Vol. 2, pp 769 – 773, August 2007
- [54] The MathWorksTM, <http://www.mathworks.com/products/matlab/description1.html>
- [55] H. Shaalan, J. Thompson, R. Broadwater, M. Ellis and H. Ng, "Distribution Engineering Tool Features a Flexible Framework," IEEE Transactions on Computer Applications in Power, Vol. 8, Issue 3, pp 21 – 24, July 1995
- [56] H. E. Shaalan, J. Thompson, R. P. Broadwater and H. Ng, "DEWorkstation: A Comprehensive Distribution Modeling Tool," IEEE Proceedings of Transmission and Distribution Conference, pp 7- 12, September 1996

References

- [57] R. Broadwater, J. Thompson, M. Ellis, H. Ng, N. Singh and D. Loyd, "Application Programmer Interface for the EPRI Distributed Engineering Workstation," Proceedings of IEEE Transactions on Power Systems, Vol. 10, No. 1, pp. 499 – 505, February 1995
- [58] Java Agent Development Framework, <http://jade.tilab.com/>
- [59] F. Bellifemine, G. Caire and D. Greenwood, "Developing Multi-Agent Systems with JADE," John Wiley & Sons, Ltd., 2004
- [60] Wikipedia, http://en.wikipedia.org/wiki/Graph_theory
- [61] C. Rehtanz, "Autonomous Systems and Intelligent Agents in Power System Control and Operation", Springer, 2003
- [62] K. L. Butler-Purry, N. D. R. Sarma and I. V. Hicks, "Service Restoration in Naval Shipboard Power Systems," IEEE Proceedings on Generation, Transmission and Distribution, Vol. 151, No. 1, pp 95 – 102, January 2004
- [63] P. Tulpule, "Multiagent Approach for Power System Reconfiguration," M. S. Thesis, West Virginia University, 2007