Journal of ELECTRICAL ENGINEERING, VOL. 54, NO. 11-12, 2003, 316–319

COMMUNICATIONS

DISTRIBUTED MOBILE AGENT MODEL FOR MULTIAREA POWER SYSTEM ON–LINE ECONOMIC LOAD DISPATCH

Kannan Nithiyananthan — Velimuthu Ramachandran *

The main objective of this paper is to construct a mobile agent model in a distributed environment through which the economic load dispatch (ELD) solutions of multiarea power systems can be monitored and controlled. In the proposed model for ELD, a mobile agent is created which enables the mobility of the economic dispatch executable code to the neighboring power system client host. A single server (main container) /multi client (sub container) based architecture which enables the power system clients to access the economic load dispatch mobile agent at its own framework with their respective data and to get the economic load dispatch solutions. A distributed agent model has been implemented in such a way that for every specific period of time, the ELD agent migrates from one power system client to another to obtain the system data and the optimized economic load dispatch solutions with power loss have been sent back to the respective clients in a heterogeneous environment. The ELD agent creates a new thread of control for every clients request and hence complete distributed environment is exploited.

K e y w o r d s: economic load dispatch, multi-area power systems, monitoring, automated control, mobile agent

1 INTRODUCTION

Economic dispatch problem of power systems is to determine the optimal combination of power outputs for the all-generating units, which minimizes the total fuel cost while satisfying the constraints. The Power System economic load dispatch solution obtained through conventional client-server architecture is complicated, memory management is difficult, source code is bulky, and exception-handling mechanism is not so easy. In the conventional power system operation and control, it is assumed that the information required for monitoring and controlling of power systems is centrally available and all computations are to be done sequentially at a single location [1]. With respect to sequential computation, the server has to be loaded every time for each clients request and the time taken to deliver the economic load dispatch solution is also comparatively high [3, 4, 5].

This paper outlines a new methodology for solving economic load dispatch analysis in a distributed environment. Mobile agent based client-server architecture overcomes the difficulties associated with sequential computation and it can be easily implemented.

The mobile agent framework provides a number of advantages including the saving of network bandwidth and increasing of the overall performance by allowing the ELD agent to process the power system data in the client machine itself. The proposed agent framework supports the asynchronous processing that makes ELD agent to fulfill a given task without the need to have a permanent connection from the client to a main container. The proposed architecture makes the economic dispatch application as more fault tolerant where network failure can influence only the migration of an ELD agent as the rest of the process is then done locally on the same node.

2 DISTRIBUTED AGENT MODEL FOR AUTOMATED ECONOMIC LOAD DISPATCH SOLUTION

In this present work, a distributed environment has been set up using Java Agent Development Framework [6] based mobile agent framework to estimate and to monitor economic load dispatch solutions for different subsystems of an integrated power system. Each sub system has been considered as a power system client and hence multi power system clients – single ELD agent model is implemented. In this model the tie line power flow for each area is assumed to be constant and tie line power flow is treated as load for each sub system. These power system clients are interconnected in a network where ELD agent always on the move as shown in Fig. 1. A client computer basically does the distributed power system monitoring through a GUI for every specific period of time and frequently exchanges data with the ELD agent. The ELD agent does the economic load dispatch computation and then distributes the results. Chronologically the ELD mobile agent should start the process first in the main container (host1), so that it can take the initiative to set up a connection link. The mobile agent object obtains the necessary data from the container and responds back to them with the economic dispatch solutions. Then it migrates to the next host and it serves the economic dispatch executable code. This total process of migration

* Center for Professional Development Education College of Engineering, Guindy, Anna University, Chennai–25, India
E-mail: knithiyananthan@tutie.com, rama@annauniv.edu.

ISSN 1335-3632

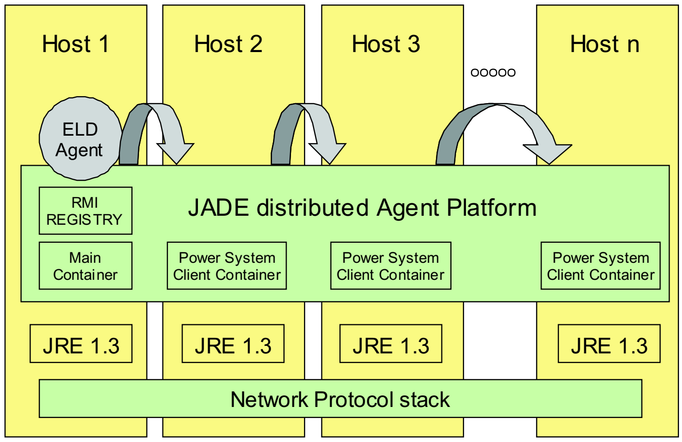

Fig. 1. ELD mobile agent based Architecture for Economic load dispatch

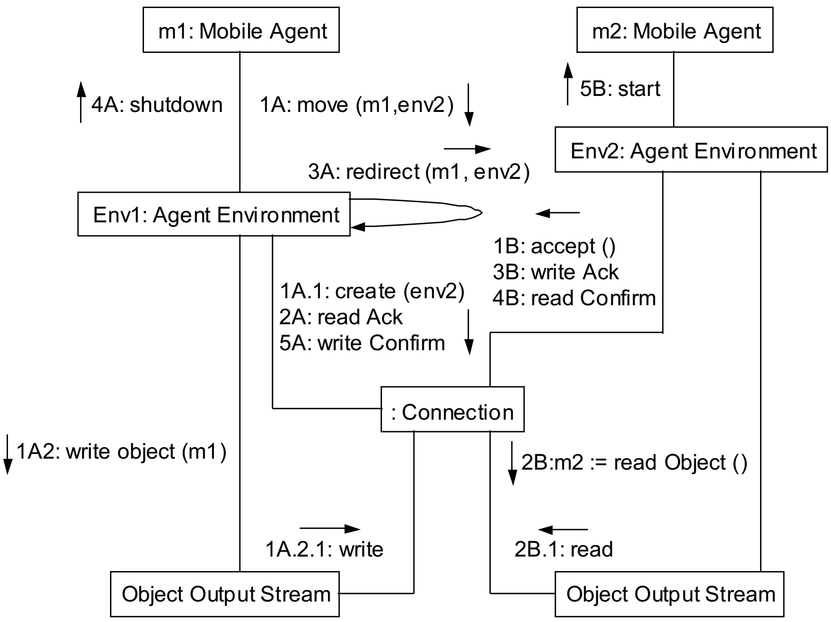

Fig. 2. ELD Mobile Agent Migration

of ELD agent from one host to another made for every specific time interval. Transaction of ELD agent from different host takes place several times, so the possibilities of the occurrence of errors and security for the power system data may be high. Hence it must be handled properly and it takes care by JADE distributed environment automatically.

2.1 Mobile Agent Flow Model [2]

In this proposed mobile agent is made to migrate from one container to another as shown in Fig. 2. At first instance mobile agent calls one of its environments methods requesting its migration to new environment env2 and the current environment creates a network connection with the new environment. Once the new environment accepts the network connection from the current environment and then current environment uses an ObjectOutputStream to serialize the mobile agent to a stream of bytes and send the stream to the new environment. At another instance the new environment calls the readObject method of an ObjectOutputStream, so it can reconstruct the ELD mobile agent from the byte of stream. After this instance the new environment sends the acknowledgement to the current environment that it received the agent. Since the instance of the mobile agent in the new environment is not yet active, the current environment tells the instance of mobile agent to shutdown and its sends a confirmation to the new environment that the old instance has shutdown. Once this shutdown status confirmation is received

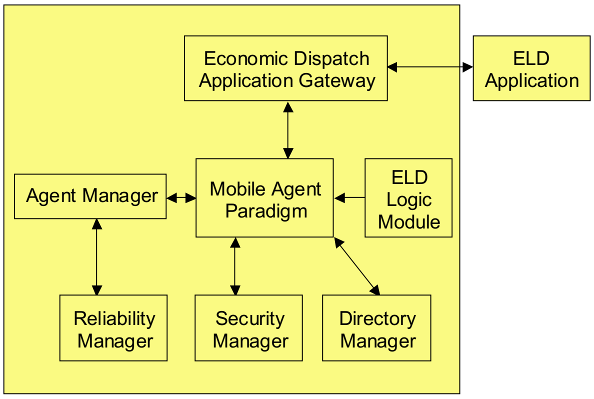

Fig. 3. ELD Mobile agent Paradigm

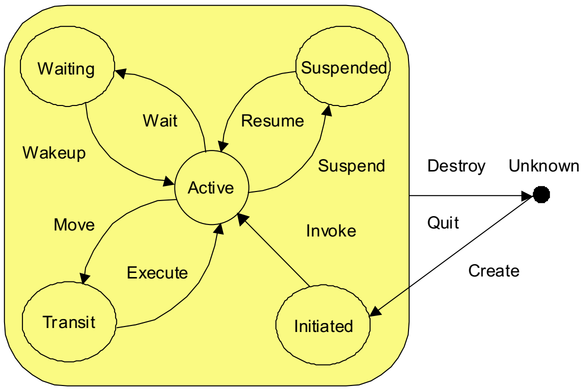

Fig. 4. Mobile Agent Life Cycle

the new environment allows the mobile agent instance to be active. This collaboration of work makes the mobile agent to migrate to new environment.

2.1 Mobile Agent Paradigm

The major components of the mobile agent architecture are an agent manager, a security manager, an application manager and a directory manager as shown in Fig. 3. All these major components mentioned have individual responsibilities to perform. The agent manager receives ELD agent for execution on the local host and also sends agents to remote power system clients and before transportation of agent, the agent manager serializes the agent and its state. It then passes the serialized form to its counterpart on the destination host. In a highly reliable architecture it actually passes the agent to the reliability manager, which ensures that the agent manager on the remote host receives the agent. Upon receipt of an agent, the agent manager reconstructs the agent and the objects it references and then creates its execution context.

The Security manger authenticates the agent before it is allowed to execute. Thereafter, the Java virtual machine automatically invokes the security manager to authorize any operations using system resources. ELD mobile agents may use directory manager to identify the location of a next neighboring host. An arriving mobile agent accesses to resident servers such as database servers through this gateway. ELD Logic encapsulates the agent behavior and logic of economic dispatch application. This architecture prescribes that an agent is a composite java object that includes mobility, persistence and can communicate with other hosts. During its life cycle a mobile agent receives various kinds of events in response to its actions. For instance, if it moves to another host, a mobility event occurs just before and after the migration and corresponding call - back mechanism is invoked. In this way, each event gives to the agent the opportunity to determine how to react. A programmer implements a mobile agent filling its call - back methods such as before Dispatch and after Arrival as appropriate.

3 ELD AGENT LIFE CYCLE

An ELD mobile agent can be in one of several states, according to agent Platform Life Cycle in FIPA specification; these are represented by some constants in Agent class in the JADE environment as shown in Fig. 4. The states are

- **AP_INITIATED**: the ELD agent object is built, but has not registered itself yet with the Agent Management System, has neither a name nor an address and cannot communicate with other agents.
- **AP_ACTIVE**: the ELD agent object is registered with the AMS, has a regular name and address and can access all the various JADE features.
- **AP_SUSPENDED**: the ELD agent object is currently stopped. Its internal thread is suspended and no agent behavior is being executed.
- **AP_WAITING**: the Agent object is blocked, waiting for message. Its internal thread is sleeping on a Java monitor and will wake up when some condition is met (typically when a message arrives).
- **AP_DELETED**: the ELD agent is definitely dead. The internal thread has terminated its execution and the Agent is no more registered with the AMS.
- **AP_TRANSIT**: a mobile agent enters this state while it is migrating to the new location. The system continues to buffer messages that will then be sent to its new location.
- **AP_GONE**: JADE internally uses this state when a mobile agent has migrated to a new location and has a stable state.

The above distributed mobile agent model has been implemented in Windows NT

The above GUI shows the economic load dispatch solution for a specific 3-generator bus power system client. In this GUI, once start button event is triggered ELD agent will receive the power system data and starts the computation of economic dispatch solution. Once the economic dispatch solution is displayed through GUI move button event has to trigger by selecting the container or environment in the available locations window where the ELD

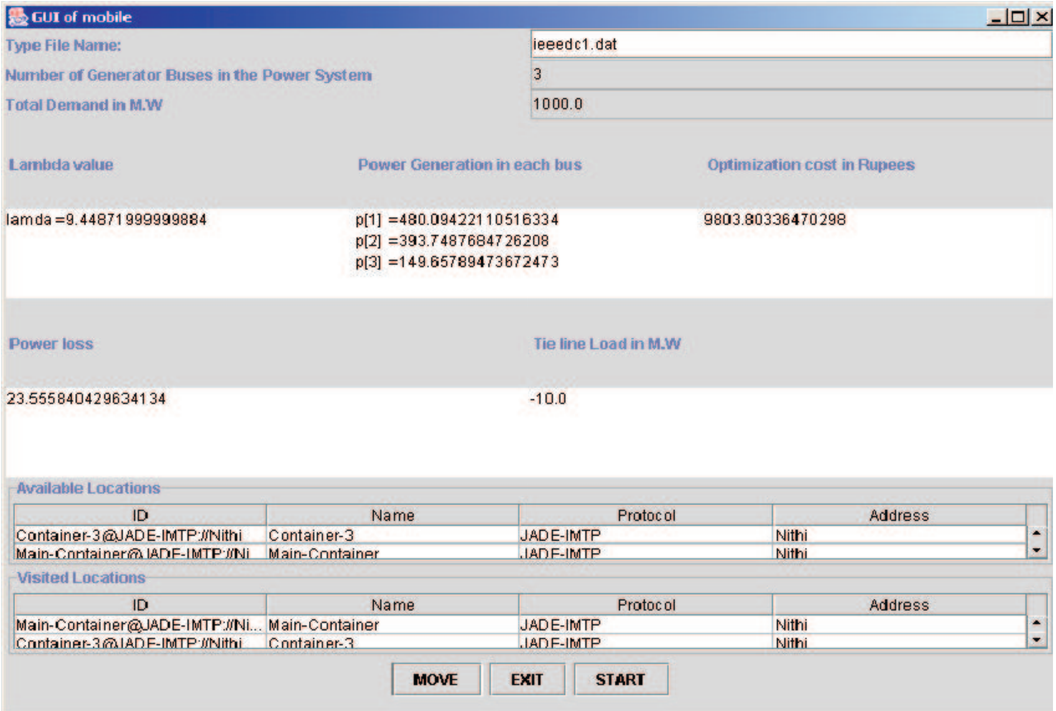

Fig. 5. GUI with economic load dispatch solution

mobile agent has to move. Using this approach, different power system clients can monitor continuous updated optimized economic load dispatch solutions at regular time intervals.

5 CONCLUSION

An effective distributed agent model has been developed to monitor the economic load dispatch of multi-area power systems. It has been tried out in overcoming the overheads associated with sequential power system economic load dispatch computation through this model. Although, client-server architecture for economic load dispatch solution is very well established, the value of this study lies in that it emphasizes a unique mology based on mobile agent model to serve a large number of clients in a distributed power system environment, across various JADE platforms based on communication between virtual machines. A practical implementation of this approach suggested in this paper was assessed based on 3, 6, 9 and 13 bus sample systems. Accordingly the proposed model can be implemented for large power systems network spread over geographically apart.

References

[1] WOODA. J.—WOOLLENBERGB. F.: Power Generation, Operation and Control, John Wiley & Sons, New York, 1984.

[2] GRAND. M: Java Enterprise Design Patterns Volume III, Wiley Computer Publishing, 2002.

[3] BANDYOPANDHYAY, G.—SENGUPTHA, I.—SAHA, T. N.: Use of Client-Server Model in Power System Load Flow Computation, IE(I) Journal-Electrical **79**, February 1999, 199–203.

[4] HART, D. G.—UY, D.—NORTHCOTE-GREEN, J.—LAPLACE, Cr.—NOVOSEL, D.: Automated Solutions for Distribution Feeders, IEEE Computer Applications in Power, October 2000, 25–30.

[5] QIU, B.—GOOI, H. B.: Web based SCADA Display Systems (WSDS) For Access via Internet, IEEE Transactions on Power Systems **15** No. 2, May 2000, 681–686.

[6] BELLIFIMINE, F.—CAIRE, G.—TRUCCO, T.—RIMASSA, G.: JADE Programmers Guide.

Received 16 July 2003

Kannan Nithiyananthan is currently working as a Teaching/Research Associate in the Department of Electrical and Electronics Engineering, College of Engineering, Guindy, Anna University, INDIA. He had received his Bachelor of Engineering (Electrical and Electronics Engineering) and Master of Engineering (Power System Engineering) from Faculty of Engineering and Technology, Annamalai University, Chidambaram, INDIA, in 1998 and 2000 respectively. His research interests include Power Systems analysis and modelling, distributed computing and Internet Technologies.

Velimuthu Ramachandran is currently a Professor of Computer Science and Engineering in College of Engineering, Guindy, Anna University, INDIA. He has received his Masters of Engineering and PhD in Electrical Engineering from College of Engineering, Guindy, Anna University, Chennai, INDIA. His research interests include Power Systems reliability engineering, network security, component technologies and soft computing.