Simulating Electricity Markets With Java

Y. C. Lam Felix F. Wu
Department of Electrical and Electronic Engineering
The University of Hong Kong

Abstract: In recent years, the power systems of many countries have been restructured. Different models of electricity market emerged. A simulator that can serve as an educational tool, an evaluation tool, and a training tool would especially be useful for countries that are contemplating restructuring. A power market has significant requirements on platform independent and interactive capabilities. In this paper, the simulation of electricity markets using Java technology is explored. Multi-agent system template is used.

Keywords: electricity market, deregulation, Java, multi-agent system, simulator

1. INTRODUCTION

The electric power industry is undergoing a profound change. The industry in many countries have been completely restructured, e.g. England and Wales, Norway, Argentina and Chile, the United States, etc. Competition is introduced to electricity generation. Transmission and distribution systems have been opened. Customers have more choices. Economical efficiency and improved electricity services are expected to follow.

Different models of restructuring have been adopted in different countries. So far there is no single model that can claim to be ideal. Those who have started the restructuring process have undergone modifications and fine tuning from time to time. It is like ongoing real-time experiments. That leaves it a difficult task to decide the proper form of a competitive electric industry for many developing countries, especially in Asia, who are under pressure from both foreign investors and domestic consumers to follow suit of the Western countries. Developing countries cannot afford to go through real-time experiments. Computer simulation provides an alternative. Simulation can be used to assess the effects of adopting certain model to local conditions, as well as industry and consumer responses. Through repeated applications, hidden benefits and potential damages to the society may be revealed. It would assist in the decision to adopt or to modify an existing model to suit for their needs and ensure a smooth transition from the monopolistic industry to a competitive paradigm.

Simulators are effective learning and training tools. As the new structure and the operating mechanism of the power system is very different from the past, simulations provide hands-on experiences to learn the intricacies of the new system. After a new model is adopted, a market simulator will be useful to allow new participants to the market to familiarize with market structure and market operating mechanism, and to help participants to develop strategies towards different market conditions.

The requirements of a simulator reflect the requirements of the system it simulates. In a restructured power system, market participants are dispersed in locations and communication between them is of utmost importance for running the market smoothly. The Internet is the natural medium of communication as it is fast and convenient. Agent technologies can be a means to help in the coordination and negotiation between the market participants. In this paper, the design of a market simulator of the restructured power systems is presented. The simulator is built on agent technology using Java language. Java is now more likely to be described as a technology because of the vast development. The use of Java enables market participants, who are using different types of computer machine and working platform, to communicate. In this way, the communication and coordination between market participants in the deregulated power system can be facilitated. The objective of this paper is to introduce the advantages of the use of Java language and multi-agent system on simulating the electricity market.

2. ELECTRICITY MARKET

Different countries adopt different models of electricity market. The operating mechanisms of different electricity markets would vary from others. The centralized pool model was introduced in England and Wales and followed by the bilateral model of Norway. California gone through a heated debate and finally designed a system that incorporates both pool operation and bilateral contracts. Other variations have since been introduced in other parts of the country. The initial design of our simulator uses the California system [5,6,10,11] as a foundation due to its comprehensiveness. The California system provides sufficient information to guide the requirement analysis of a market simulator. The simulator is, of course, designed to be flexible, extendable and modifiable, so that it can be changed to simulate different industry models.

2.1. Electricity market in California

For the purpose of requirement analysis, we focus on (1) market participants, (2) their own decision-making requirements, and (3) their interactions with other participants.

2.2. *Market participants and distributed decision making*

The California system features many participants in the overall market, notable among them are:

- *Independent System Operator (ISO)*

It controls and operates the transmission system, schedules the delivery of electric power supplies to ensure the actual demand is met with sufficient power supplies. It also procures ancillary services, administers congestion management, billing and settlement system.

- *Power Exchange (PX)*

It provides a competitive energy auction opens to all supplies and market purchases. It determines the price of electricity on an hourly basis for the Day-Ahead and Hour-Ahead markets according to supply bids and demand bids. It performs clearing and settlement functions.

- *Schedule Coordinator (SC)*

It submits balanced schedule to the ISO. It also settles with generators and retailers. PX is a special case of a schedule coordinator.

- *Generator*

It submits supply bids of power to the PX or other SCs and generating electricity under agreement. It may also bid ancillary services into the ISO or self-provide through a SC.

- *Customer*

It can choose direct access via local utility retailers, power marketers or generators.

- *Utility Distribution Company (UDC)*

It is a special case of customer.

- *Retailer, Broker and Aggregator*

2.3. *Interactions among market participants.*

The market structure and operating mechanism can be represented by the Figure 1. The interactions among market participants are indicated in the figure.

3. MAIN REQUIREMENTS OF A MARKET SIMULATOR

The participants of an electricity market are geographically dispersed and make independent decisions. Different tools are used by different participants. Market function requires interactions among participants. Interactions among participants are also necessary to ensure reliable operation of the power system. Participants may use different computing platforms. Thus, the four major requirements of a market simulator are:

- Distributed decision making
- Communications among participants
- Platform-independent computing
- Interface with various software tools

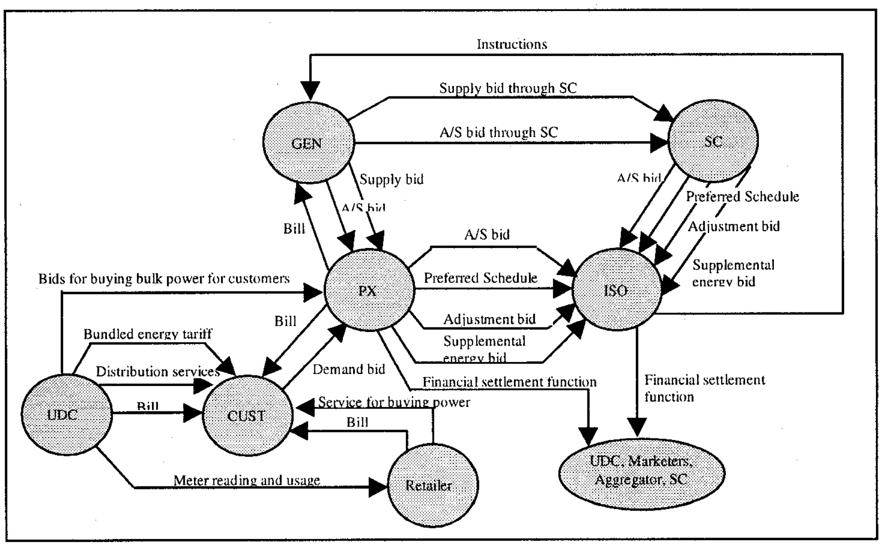

Figure 1
The overall picture of the functions of the market participants and the communication between market participants in the California power market.

We propose to use Java technology [8] and Multi-agent systems [2,3] as the foundation to build the market simulator. Java technology is designed for Internet and is suitable for applications in simulation of electricity markets, where market participants are dispersed in location and large amount of data transfer and manipulation are required. Multi-agent system provides a template for distributed decision-making by independent agents. A brief introduction to Java technology and multi-agent systems will follow.

4. JAVA TECHNOLOGY

Java is designed for use on the network, which requires software to be portable and modular and is described as platform-independent. It works on different hardware architectures and applications can be deployed into different network environments. The compiled Java programs can be run on different machines with different environment, provided that there is a Java-capable browser or Java Virtual Machine (JVM) in each machine. Java Virtual Machine (JVM) is the architecture-neutral and portable language platform of Java. It is the specification of an abstract machine for which Java language compilers can generate code. The compiled Java program, in the form of bytecodes, is interpreted on any platform by its own JVM. Bytecode is an architecture neutral intermediate format for transporting codes to different hardware and software platform. JVM reads the stream of bytecode as a sequence of instructions for performing different applications.

Java language is an object-oriented programming language. It defines data as objects and the methods that may be applied to those objects [1]. Java, however, supports only single inheritance, unlike many object-oriented programming languages, which support multiple inheritances. Single inheritance avoids the problem of a class inheriting classes whose behaviour are contradictory or mutually exclusively. Interfaces may be designed to accommodate situations where multiple inheritance is called for. Interfaces permit type substitution in a way, which is similar to multiple inheritance, but are not a substitute for multiple inheritance. They do not promote the same level of code reuse.

Java applets are Java programs that can be included in an HTML (Hypertext Markup Language) page and it can be run all over the World Wide Web. When a Java-capable browser is used to load an HTML document containing a reference to an applet, the applet's code is transferred to one's local system and executed by the browser.

There is a set of Application Programming Interface (API) in the Java development kit developed by Sun® Microsystems. An API is a set of classes, which is designed to assist program developers in developing their own applications. There are other APIs available or developed by developers for different applications, in addition to the core APIs in the development kit.

5. MULTI-AGENT SYSTEM (MAS)

The term "agent" appears in many research areas. "Agent" here means "software agent". That is, it is a software program, which can perform some task as implemented by the program developer. The use of software agent is to remove the burdens of human beings from those tedious and repeating tasks. For those software agents, which can perform interactive task tailored for users' needs and they have special capabilities such as learning, decision making and communications, they are called "intelligent agents". There is no formal definition for software agents. However, usually, software agents have some of these characteristics, (a) autonomous, (b) rational, (c) intelligent, and (d) having incorporation capability.

A MAS is system of special type of agents. It focuses mainly on the coordination and communication among agents. Communication is very crucial to a MAS. Good coordination can only be achieved if there is good communication between agents as agents need to transfer plans and negotiate among themselves.

KQML (Knowledge Query and Manipulation Language) [12] is a language and protocol for exchanging information that supports agent communication. KQML supports run-time knowledge sharing among agents. It can also be used as a language for the interaction between application programs and intelligent systems, to share knowledge in support of cooperative problem solving. It is used in the multi-agent system we used for the electricity market simulator.

6. SIMULATOR FOR ELECTRICITY MARKET

The market simulator is built on Java, market participants can run on heterogeneous platforms and make use of very lightweight applets. Therefore, market participants can work on PCs or workstation on UNIX platform and communicate with each other and perform their operation according to the market operating mechanism over the Internet (Figure 2).

6.1 Structure of the simulator

The market simulator is divided into four layers. The most crucial issue between market participants in the market simulator is communications and coordination. Therefore, the basis of the simulator is a communication platform, which is provided by the multi-agent system (MAS) template. The market participants in the system will operate according to market operating mechanism. The second layer of the simulator is a set of classes for the functions of different kind of market participants and the operating mechanism.

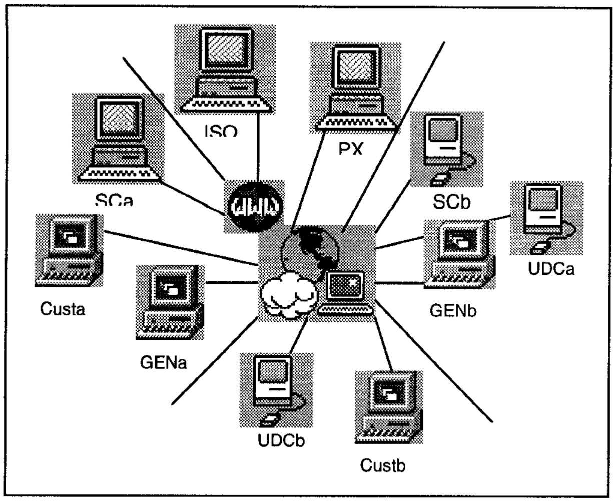

Figure 2
Market participants are connected over the Internet.

As shown in Figure 1, each block in the simulator each block represents a market participant and the interactions among them. Further, in the simulator, there are a lot of data required, e.g. network data, bid data. Therefore, each agent in the simulator should have its database to carry out the computations required. Thus, the third layer is the database layer. The data in the database should be able to reflect the market condition [4]. Therefore, the top layer in the market simulator is for putting market simulating algorithms. The structure of the simulator is shown in Figure 3.

6.2 MAS Template

Each type of market participants is represented as an agent in the simulator. The simulator is built on top of a MAS Template, JATLite [9]. The MAS Template is called JATLite (Java Agent Template, Lite). It is implemented by the Stanford University. It is a package of programs written in Java. It allows program developers to create software agent to communicate over the Internet. JATLite provides a basic infrastructure in which agents can register with Agent Message Router facilitator using names and passwords. Agents can connect and disconnect, send and receive messages, transfer files among themselves. The MAS Template is responsible for the communications among agents. The specific functions of each agent and computation algorithms are implemented according to the need of the simulator.

6.3 Database

Database is another important part of the simulator. As there is large amount of data to be transferred among market participants or used in the computations done by them, a database is needed. A database allows data to be specifically organized. In the simulator, the database contains data about the market data and network data. For market data, it includes supply bid data, demand bid data, adjustment bid data, market clearing price, market clearing quantity, ... etc. For network data, it comprises bus data,

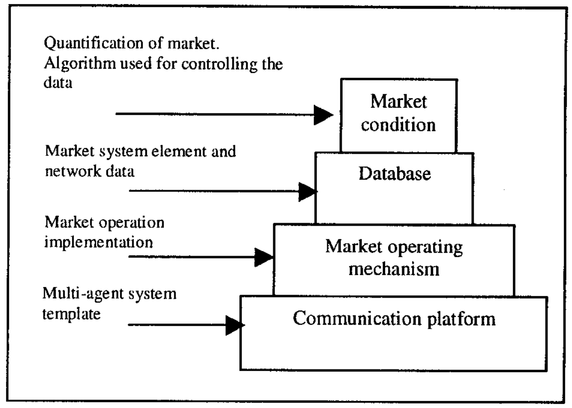

Figure 3
The structure of the electricity market simulator.

line data, voltage limit, line limit, ... etc. In order to access the database from the Java programs, JDBC (Java Database Connectivity) is used [7]. JDBC is a Java API which is used for executing SQL (Structured Query Language) statements from Java programs, which is an agent in our case. JDBC first establishes a connection with the database. In our simulator, mSQL [13] database is used. From the Java applets, JDBC is used as a bridge to connect to the database. Therefore, data can be retrieved and being processed by the agents.

6.4 Agent Computations

With the communication platform implemented, the agent actions should be considered. In the electricity market simulator, some agents, e.g. PX and ISO, have tasks that require extensive computations. For example, the ISO has to run loadflow for checking the feasibility of the Initial Preferred Schedule and the preferred schedules after congestion management. Furthermore, the ISO should run optimal power flow for the congestion management. Therefore, for simulating the operation of different processes in the electricity market and the operation of the power system, a lot of calculations and manipulations are involved. In the simulator, these computation algorithms and calculation methods certainly should not be implemented again with Java from scratch. For instance, loadflow calculation can be done by using programs, which have been implemented in C++, or some other number crunching codes. In order to save time and gain access to these special capabilities, which are not provided in the Java class library, JNI (Java Native Interface) is used. JNI is another Java API. It provides the capability to agents (Java applets) to invoke native methods. Native methods are bits of platform-specific executable code contained in libraries or DLLs. The simulator makes use of this API to invoke, for example, the Power System Toolbox from Matlab® for loadflow calculations, as well as the use of the Optimization Toolbox in solving optimal power flow during congestion management, in our prototype implementation.

The results got from the computations using native methods are sent to the participants for further agent actions.

7. CONCLUSION

Software capable of simulating intricate workings of new electricity markets provides a useful learning, evaluation and training tool. Such a simulator should be built on an environment that is conducive to distributed decision-making, efficient communications, easy interface with user's own analysis programs, and should be platform independent. We propose to use Java technology on Internet and Multi-agent system framework to build an electricity market simulator. Java virtual machine makes it platform independent. A multi-agent system template in the Java environment called JATLite is used as the template for interface and communications between agents, which are market participants. A Java API, JDBC, is used for agent-database interface, i.e., executing SQL statements from Java programs. Another Java API, JNI, is used for agent to invoke native methods, such as power system analysis or market analysis programs. A prototype implementation of the simulator using the California system as an example has been completed. The results will be presented.

8. ACKNOWLEDGEMENT

The research is supported by the Hong Kong Research Grant Council and US National Science Foundation.

9. REFERENCE

[1] A. F. Neyer, F. F. Wu, K. Imhof, "Object-oriented programming for flexible software: Example of a load flow", IEEE Transactions of Power Systems, Vol. 5, No. 3, August 1990.

[2] F. F. Wu, Chris S.K. Yeung, A. S.Y.Poon, "Game Theoretical Multi-Agent Modelling of Coalition Formation for Multilateral Trades", IEEE Transaction on Power Systems, August 1998.

[3] J. Contreras and F. F. Wu, "Coalition Formation in Transmission Expansion Planning", to appear in IEEE Trans. Power Systems.

[4] Arnold E. Amstutz, "Computer Simulation of Competitive Market Response", The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, and London, England.

[5] Barbara R. Barkovich, Dianne V. Hawk, "Charting a new course in California", IEEE Spectrum, July 1996.

[6] "PX Primer", California Power exchange, version 3, March 1998

[7] Hamilton Cattell Fisher, "JDBC™ Database Access with Java™", Addison Wesley

[8] http://java.sun.com

[9] http://java.stanford.com

[10] http://www.caiso.com

[11] http://www.calpx.com

[12] http://www.cs.umbc.edu/kqml

[13] http://www.imaginary.com/Java

10. BIOGRAPHIES

Y. C. Lam received her B.Eng. degree in Electrical Energy Systems Engineering from the Department of Electrical and Electronic Engineering of the University of Hong Kong. She is now a M.Phil. candidate in the University.

Felix F. Wu is currently serving as the Pro Vice Chancellor and Professor of Electrical Engineering at the University of Hong Kong, on leave from the University of California, Berkeley, where he is a Professor of Electrical Engineering and Computer Sciences. Prof. Wu was elected Fellow of IEEE in 1989 for "contributions to theory and computation methods in power system planning and operation".