An Effective Distributed Model for Power System Transient Stability Analysis

Balasingh Moses MUTHU[1], Ramachandran VEILUMUTHU[2], Lakshmi PONNUSAMY[3]
[1]*Department of Electrical Engineering, Anna University Tiruchirappalli, Tamilnadu, 620024, India*
[2]*Department of Information Science and Technology, Anna University, Chennai, 600025, India*
[3]*Department of Electrical and Electronics Engineering, Anna University, Chennai, 600025, India*
moses@tau.edu.in, rama@annauniv.edu, p_lakshmi@annauniv.edu

***Abstract*— The modern power systems consist of many interconnected synchronous generators having different inertia constants, connected with large transmission network and ever increasing demand for power exchange. The size of the power system grows exponentially due to increase in power demand. The data required for various power system applications have been stored in different formats in a heterogeneous environment. The power system applications themselves have been developed and deployed in different platforms and language paradigms. Interoperability between power system applications becomes a major issue because of the heterogeneous nature. The main aim of the paper is to develop a generalized distributed model for carrying out power system stability analysis. The more flexible and loosely coupled JAX-RPC model has been developed for representing transient stability analysis in large interconnected power systems. The proposed model includes Pre-Fault, During-Fault, Post-Fault and Swing Curve services which are accessible to the remote power system clients when the system is subjected to large disturbances. A generalized XML based model for data representation has also been proposed for exchanging data in order to enhance the interoperability between legacy power system applications. The performance measure, Round Trip Time (RTT) is estimated for different power systems using the proposed JAX-RPC model and compared with the results obtained using traditional client-server and Java RMI models.**

***Index Terms*— JAX-RPC, Power Systems Stability, SOAP, WSDL, XML**

I. INTRODUCTION

The existing power system operations are primarily desktop applications and implemented in parallel processing super computers. The simulation package for analysing the power system stability problems are programmed in FORTRAN or C. Later, these paradigms have been replaced using object oriented and component based technologies. In these technologies, the power system data and operations are encapsulated as objects that can interact with each other by sending messages between them to solve power system applications.

To monitor the power system operations, Supervisory Control and Data Acquisition (SCADA) and Energy Management System (EMS) are being developed. The systems provided by different power sector vendors run on different hardware and software platforms. The conventional client-server architecture for power system analysis is complicated, memory management is difficult, source code is bulky, and exception-handling mechanism is not so easy.

Monika [1] presented parallel implementation of the transient stability problem on Cluster of Workstations. All data partitioning and parallelization strategies are explored and classified using parallel-in-space approaches. Ramesh [2] described the emerging role of distributed computing for on-line Energy Management System (EMS) applications in power systems. He presented remarkable developments towards standardization in both hardware and software for distributed computing.

Hong Chen [3] developed a Web-based computing, which is based on Internet protocol, distributed processing and Java paradigm for the analysis of large scale interconnected power systems. Kwok-Hong Mak [4] illustrated the significant benefits in migrating SCADA systems to TCP / IP and Ethernet networking. Many power system operators already have the technical infrastructure, capability and capacity to develop a successful migration of SCADA to TCP / IP networking infrastructure. Chen [5] had an attempt to create a completely Web-based, platform-independent, power system simulation package with various analyses distributed in a clustered environment. Furthermore, it was stated that new software technologies will be incorporated as they become widely accepted, such as the eXtensible Markup Language (XML) to be used as the common format for interchanging data between the tiers. Keinosuke Matsumoto [6] proposed the communication network model of power trading systems using CORBA and Java RMI that can flexibly correspond to various network environments. The response time of these systems are more compared with other conventional systems.

Quirino Morante [7] proposed a distributed architecture based on Web and Grid computing for Power Systems Analysis. The proposed grid middleware platform uses a broker system for reserving on-demand computational resources. The proposed architecture does not support the integration with other power systems applications. Chen [8] has developed a CORBA based power quality monitoring system for sharing the information between different applications operating in different phases of power quality measurements and analyzes. This approach can only be successful if the interfaces are accepted by the standardization bodies such as IEEE and IEC, and they are universally adopted by all vendors. Nithiyananthan [9], [10] developed an effective RMI based distributed models for monitoring the load flow and economic load dispatch of multi area power systems. They have been tried out in overcoming the overheads associated with sequential power system economic load dispatch computation. Afaneen [11]

Digital Object Identifier 10.4316/AECE.2011.03012

developed multi-agent technology for monitoring and controlling the electrical network to enhance power system transient stability. The instability of the power system after a three phase fault at bus bar was identified using the prediction agent and the stability of system is established by applying the control agents. Shekhar [12] demonstrated the efficient application of Web services in the SCADA systems in .NET platform using Multi Media Interface (MMI) and Wireless Application Protocol (WAP). The platform provides highly desirable features for modern executives so that they can avail the vital data at any place in the world without carrying much of computer hardware and software.

The software architecture had changed from central to distributed and distributed to Web-based, and now to Web service environment. Since the existing power system is highly complex and widely distributed, it is needed to develop service oriented open system made up of a variety of power system services operating on dissimilar platforms, so that more extensive data and applications can be shared easily and flexibly. The proposed JAX-RPC (Java API for XML based Remote Procedure Call) model for carrying out power system stability analysis will provide powerful set of features based on open standards like XML, SOAP, WSDL..

II. XML Data Representation for Transient Stability Analysis

The data exchange and information sharing are the major problems in power system applications [13]. Nowadays, Energy Management system (EMS) and Database Management System (DMS) applications are based on standardized software and hardware platform. The data have been stored in different formats and are distributed in a heterogeneous environment. In this paper, an XML based model has been proposed for solving the data exchange problem between of heterogeneous systems and it is shown in Fig. 1. Different enterprises like generation, transmission and distribution have many isolated server and information management systems which are constructed with different core technologies make the data exchange complicated between the power system applications. Under these conditions, power sector needs a simple, effective and inexpensive way to realize the data sharing and exchange between applications. IEEE recommended common data format for exchanging power system data is limited to static data and not suitable for storing dynamic information [14].

Rapidly growing power sector environment changes their data structure or adds new data structure whenever new participants join in the power network. The data exchange must have a protocol which makes the data meaningful for each power system operation. The XMLised representation of power system data offers reliable data exchange between legacy power system applications. All the data required for stability analysis are stored in Oracle database. The Stability Service Data Objects (SSDO) is a container for information designed to promote open standards and interoperability.

Certain data are common to all power system applications. All clients in the interconnected power systems have to know the general specification of the system. The base value of the system, number of buses, number of generators, transmission lines, transformers, acceleration factor and tolerance value are classified as general data.

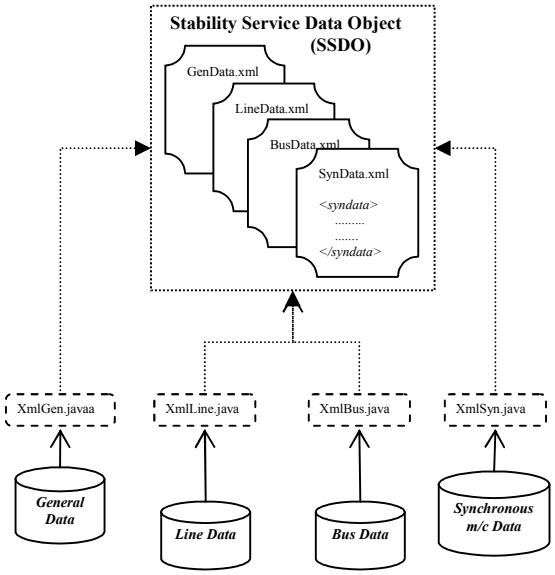

Figure 1. Conversion of Power System Data into XML document

The XML document generated from Oracle database for representing general data for power system analysis is as follows:

```
<general>
 <basemva>Base value of the System</basemva>
 <nb>Number of Buses</nb>
 <ng>Number of Generators</ng>
 <nl>Number of Lines</nl>
 <nt>Number of Transformers</nt>
 <alpha>Acceleration factor</alpha>
 <tolerance>Tolerance  value</tolerance>
</general>
```

The power system line data includes transformer line rating which is essential for reliable planning and operation of the interconnected systems. This rating incorporates values for resistance, reactance, off line charging admittance and acceptable electrical loading on equipment before, during and after system disturbances. The way in which the lines are connected between the buses is also stored in the line data. The transmission line data would have to be updated based on weather and seasonal conditions. The XML document obtained from Oracle database for representing line data required for stability analysis is as follows:

```
<linedata>
  <sb>Sending Bus</sb>
  <rb>Receiving bus</rb>
  <r>Resistance of the line</r>
  <x>Reactance of the line</x>
  <b>Off line Charging admittance</b>
  <rating> Maximum rating of the line</rating>
</linedata>
```

The generation capacity and the consumer demand are major factors for analysing the stability of the power systems [15]. The power system bus data includes magnitude and phase angle of voltages, real and reactive power flow on the interconnected systems and the rating of shunt capacitor. The operating limits of the buses have to be represented for the violations of the power system

operations.

The XML document for representing bus data required for stability analysis is as follows:

```
<busdata>
 <slackbus>
    <voltage>Voltage at the bus</voltage>
    <angle>Load angle of the bus</angle>
 </slackbus>
 <generatorbus>
    <mw>Real Power</mw>
    <voltage>Generating Voltage</voltage>
 </generatorbus>
 <loadbus>
    <mw>Real Power</mw>
    <mvar>Reactive Power</mvar>
 </loadbus>
 <limits unit=mvar>
    <max>Maximum Réactive Power
                              Limit</max>
    <min>Minimum Réactive Power
                              Limit</min>
 </limits>
</busdata>
```

The synchronous machine data are essential to ensure stable operation of the power systems. The generating capability to meet the projected demands is the most important factor in the transient stability analysis. The synchronous machine data emphasize the values of synchronous reactance, synchronous speed, moment of inertia, number of poles and damping factor. The XML document obtained for representing synchronous machine data is as follows:

```
<machinedata>
    <xs>Synchronous reactance</xs>
    <w>Synchronous speed</w>
    <n>Number of poles </n>
    <j>Moment of Inertia</j>
    <d>Damping factor</d>
</machinedata>
```

The conversion of power system data into an XML form enables the independency of the data used by various power system applications. The proposed XMLised power system data representation model significantly reduces the engineering efforts required to integrate its data in the Web service environment. This ensures interoperability between various power system applications in a heterogeneous environment.

III. Proposed Distributed Model for Transient Stability Analysis

The power system transient stability operation is represented as a Web service in a distributed environment using JAX-RPC. The power system clients communicate with stability service endpoints for invoking the stability services in JAX-RPC model as shown in Fig. 2. The API for JAX-RPC allows the exchange of power system data in XML format. The XMLised power system data is attached with SOAP message and can be communicated to the stability services through Stability Service Descriptors (SSDs). The power system client needs only the SSDs to access the stability services.

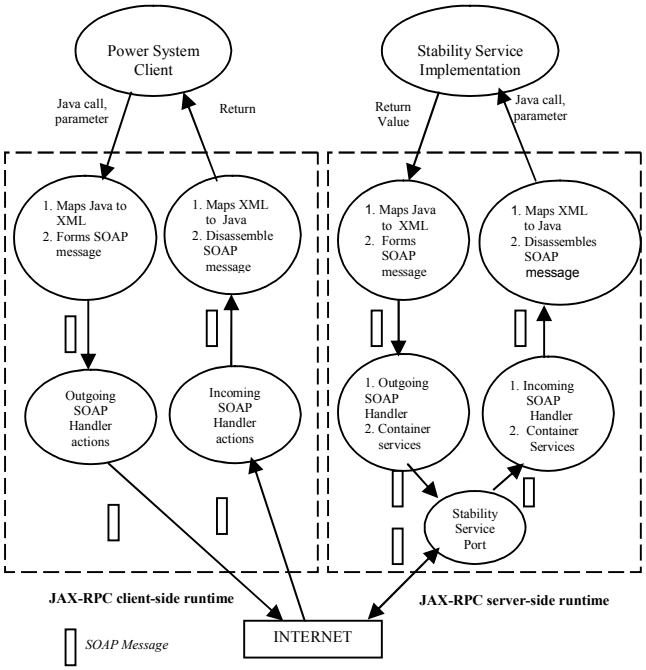

Figure 2. Implementation of power system stability services in JAX-RPC

The power system client makes a request to a particular Stability Service by invoking a Java method, along with setting up and passing the required parameters such as general data, line data, bus data and machine data and receives the appropriate response as the result of method invocation. The JAX-RPC runtime maps the Java types to standard XML types and forms a SOAP message that encapsulates the method call and parameters, then passes the SOAP message through the SOAP handlers and then to the server-side service port. The JAX-RPC enables exchange of SOAP requests and responses through an API that hides the details about the SOAP message to the power system client. The power system client can access the deployed stability services using a Stability Service Endpoint Interface (SSEI). The SSDs have a reference to the SSEI. The SSEI is a remote interface that declares the remote methods through which the power system client interacts with the stability services.

IV. Implementation of Proposed Distributed Model

The various stages involved in the implementation of proposed JAX-RPC model for Stability analysis are data representation, defining stability service endpoint interface, service configuration, service description, service mapping, service deploying and invoking.

A. Stability Service Endpoint Interface

In JAX-RPC, the stability service endpoint interface must extend the java.rmi.Remote interface. All methods implemented by this interface must throw a java.rmi.RemoteException. The method parameters must be JAX-RPC-supported Java types. The service interface provides the contract between the power system client and server as shown in Fig. 3. The power system clients can

communicate to the stability service providers through stability service endpoints.

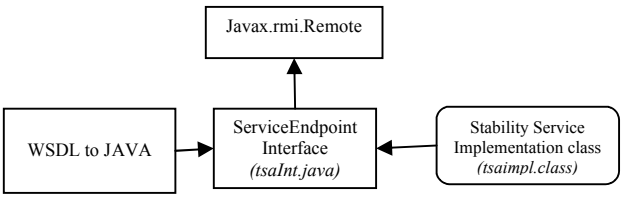

Figure 3. Stability Service Interface and Implementation

The Pre-Fault, During-Fault, Post-Fault and Swing curve services are included in the service endpoint interface. The service implementation class actually defines the methods declared in the service endpoint interface. The service endpoint interface for computing swing curve is as follows:

```
package TSARPC;
public interface tsaInt extends Remote
                // transient stability analysis    interface tsaInt
   {
     public String computeSwing( ) throws
                                   RemoteException ;
    }
```

The power system clients need not be aware of any underlying technology or programming paradigm which the service is using. The service interface encapsulates all aspects of the network protocol used for communication between clients and service provider. All other services are declared similarly.

B. Configuring the Stability Service

The configuration of power system stability services are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns="http://java.sun.com/
                          xml/ns/jax-rpc/ri/config">
      <service
          name="RPCStability"
          targetNamespace="urn:TSA"
          typeNamespace="urn:TSA"
          packageName="TSARPC">
         <interface name="TSARPC.tsaInt"/>
      </service>
</configuration>
```

This configuration file contains the information and details about the deployed stability services and metadata such as their service name (RPCStability), package name (TSARPC), namespace (TSA) and the interface (tsaInt). This configuration file is used to create the stability service descriptors and mapping information about the services in order to communicate between power system client and stability service provider.

C. Stability Service Description

The power system applications are implemented using various language paradigms. In JAX-RPC, the WSDL is used as the metadata language for defining the stability services. It describes how the service end point and client communicate with each other. WSDL is capable of describing services that are implemented using any language and deployed on any platform. It represents information about the interface and semantics of how to invoke a service. It contains the information about the data type, binding and address information for invoking the services from the service provider. The swing curve stability service is described as follows:

```
<definitions name="RPCStability"
          targetNamespace="urn:TSA"
                        xmlns:tns="urn:TSA"-->
    <types>
      <schema targetNamespace="urn:TSA"
                          xmlns:tns="urn:TSA">
      </schema>
      <complexType name="ArrayOfdouble">
      </complexType>
    </types>
    <message name="tsaInt_computeSwing"/>
      <part name="result" type="xsd:string"/>
    </message>
    <portType name="tsaInt">
      <operation name="computeSwing”>
        <input
          message="tns:tsaInt_computeSwing"/>
        <output
          message="tns:tsafacecompute
                              SwingResponse"/>
      </operation>
    </portType>
    <binding name="tsaIntBinding"
                              type="tns:tsaInt">
      <soap:binding transport=
          "http://schemas.xmlsoap.org/soap/
                                style="rpc"/>
    </binding>
    <service name="RPCStability">
      <port name="tsaIntPort"
                    binding="tns:tsaIntBinding">
      <soap:address   location="http://
              loacalhost:8080/stability"/></port>
    </service>
</definitions>
```

The service descriptor document consists of several key structural elements for describing stability service. The <definitions> element defines the name of the service as ‘RPCStability’ and declares the namespace as ‘TSA’. The <types> element defines the schema and data types that would be used to describe the stability data. The <message> element represents the name of the method to be invoked and the response type. The <portType> element provides the abstract definition of the operation (tsaInt) of the service, request and response messages. The <binding> element specifies a concrete protocol (SOAP) used for representing messages. The <service> element represents the port name (tsaIntport) and location of the services.

D. Configuring the Stability Service

In the traditional distributed environment paradigm such as RMI, the method call and the associated parameters are marshalled (i.e.) converted into a wire format while clients try to invoke a remote method. At the server end, the marshalled data has been unmarshalled (i.e.) converted back to original method call and parameters. Marshalling and unmarshalling will be applicable only to Java paradigm and confine to RMI framework. Since the existing power system applications are implemented in different language paradigms, they need a common serialization / de-serialization of information for invoking the remote services.

The common format has to be flexible and interoperable in a heterogeneous environment. In this model, all the stability service objects are translated and mapped into XML. The mapping file describes the elements like package, type, port, method, and endpoint. While invoking the stability service endpoint interface method, the method call and its parameters are mapped in to XML and sent through SOAP communication protocol. When received at the client or server end, the request / response parameters must be mapped from XML to their proper types or objects. While WSDL converts the java interface to XML form, the mapping.xml describes the conversion from XML to Java objects and vice versa.

E. Deploying the Stability Services

In the proposed model, Tomcat Web server is used for deploying the stability services. The stability services have been deployed and packaged into Servlet container as shown in Fig. 4.

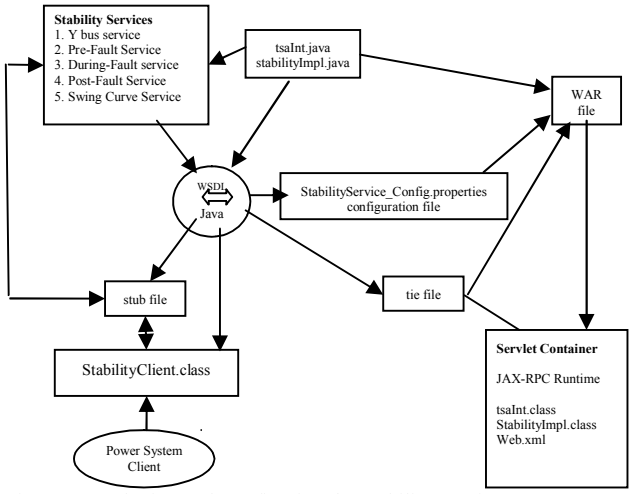

Figure 4. Deploying and Configuring the Stability Services

The service endpoint interface, service implementation class, and web.xml file, along with other generated artifacts and configuration properties file are bundled into a standard WAR file called as "stablity.war". In JAX-RPC stubs and ties are classes that enable communication between a service endpoint and client. The stub class resides on the client side (i.e.) between the client and the JAX-RPC client runtime system. The stub class is responsible for converting a request from a JAX-RPC client to a SOAP message and sending it across to the server using HTTP protocol. It also converts the response from the server, which it receives in the form of a SOAP message, to the format required by the client. These types of SOAP communication make the entire system interoperable.

Similarly, the tie class resides on the server side, between the service endpoint and the JAX-RPC runtime system. A stub is a local object that acts as a proxy for the service endpoint. The WAR file is then deployed on to the Servlet container. Successful deployment results in an URL http://192.168.1.1:8080/servlet-examples/servlet/stability (endpoint). Any power system client can use the endpoint address for accessing the stability services

F. Invoking the stability Services

An effective Dynamic Invocation Interface (DII) model is proposed to invoke the power system stability services deployed in remote server. The stability service endpoint is available to any type of client, regardless of the language or platform used. A power system client can invoke the method in the endpoint interface using Dynamic Invocation Interface model as shown in Fig. 5. The DII is a call interface that supports programmatic creation and invocation of a RPC request.

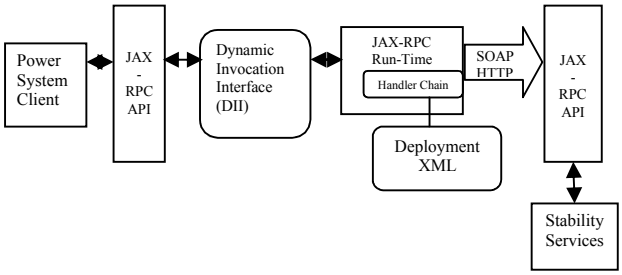

Figure 5. Invoking Stability Services

The following code delineates how the power system stability services being invoked by the power system client using Dynamic Invocation Interface.

```
ServiceFactory factory
                    =ServiceFactory.newInstance();
Service service =  factory.createService(new
                          QName(stabilityService));
QName port = new QName(tsaPort);
Call call = service.createCall(port);
call.setTargetEndpointAddress
                    ("http://localhost:8080/stability");
QNAME_TYPE_STRING = new
                          QName(NS_XSD, "String");
call.setReturnType(QNAME_TYPE_STRING);
call.setOperationName( new QName(BODY_
          NAMESPACE_VALUE,"computeSwing"));
String result = (String)call.invoke();
System.out.println(result);
```

Using DII, a client can call a service or a remote procedure on a service without knowing the exact service name or the procedure's signature ahead of time. The power system client can discover the information at runtime, and can dynamically look up the service and its remote procedures.

V. Performance Analysis

The major factor that influences the performance of the proposed distributed model is the Round Trip Time (RTT). RTT is the time that elapses between the initiations of a method invocation by the power system client until the results are returned to the client. The power system transient stability analysis is being carried out for 6, 14 and 39 bus systems. The performance measure, Round Trip Time (RTT) is estimated for different power systems using the different models such as Client / Server, Java RMI and JAX-RPC. The RMI uses the JRMP (Java Remote Method Protocol), which uses TCP/IP for communication. RMI client requires an open port to communicate with RMI server. When there is a firewall between the client and the server, RMI communication is blocked by default. This can be a major problem, particularly with applications, which require connectivity and interoperability over different LANs, each secured by its own firewall. The Java-RMI is more efficient with regard to the handling of individual packets. In an Intranet, the resulting large number of small packets may cause congestion; on the Internet, there is greater chance of packet loss. Because the methods used for performance evaluation did not do any processing, the round trip time expresses the overhead of remote method invocation. The performance analysis of the different distributed models has been carried out with respect to round trip time and the numbers of clients invoking the services and the results are shown in Fig. 6.

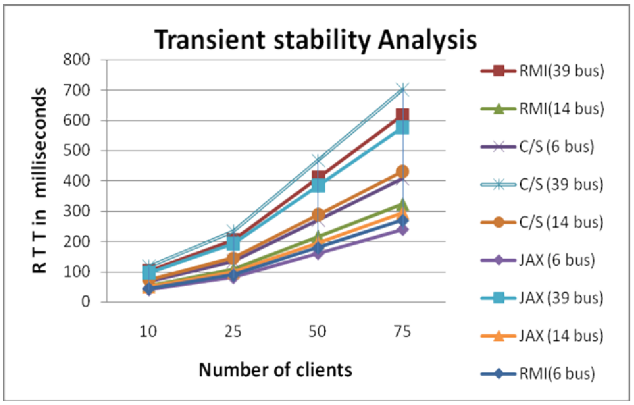

Figure 6. Number of clients vs RTT

JAX-RPC uses the XML-based SOAP communication between remote services. SOAP builds on top of existing Internet protocols, such as HTTP, FTP or SMTP. Therefore SOAP can transverse firewalls seamlessly, because firewalls will treat SOAP messages similarly as other HTTP (or FTP, SMTP) messages, respectively to the transport protocol used for SOAP messages. The JAX-RPC enables the communication over the XML-based SOAP protocol and thus enables interoperability with other Web services, which do not have to be developed in Java. The XML-based protocol for communication makes the performance of Web services will be lower than the performance of RMI. But this overhead will be inevitable due to enhancement in the interoperability between power system applications in heterogeneous environment. Each formula should occupy one line. Consecutive numbers should be marked in brackets.

VI. Conclusion

An effective distributed model has been developed for representing transient stability analysis of a large interconnected power system and tested for a sample of 6, 14 and 39 bus systems. This proposed model is scalable for any number of power system clients and the stability services can be invoked without any limitation in this Web service environment. Based on the performance analysis, the system provides excellent scalability and has a capacity to meet the huge computation requirement, which is suitable to carry out transient stability analysis for large interconnected power systems. The various power system services can be plugged into this model and the services are made available anytime and anywhere for the power system operations.

References

[1] Monika ten Bruggencate, Suresh Chalasani,"Parallel Implementations of the Power System Transient Stability Problem on Clusters of Workstations," Int. Conf. High Performance Networking and Computing, Article no 34, 1995.

[2] V.C. Ramesh, "On Distributed Computing for On-Line Power System Applications," Int. J. Electrical Power & Energy Systems, vol. 18, no. 8, pp. 527-533, 1996.

[3] Hong Chen, Claudio A. Canizares, Aajit Singh, "Web Based Computing for Power System Applications," North American Power Symposium, California, 1999.

[4] Kwok-Hong Mak and Barry Holland, "Migrating Electrical Power Network SCADA Systems to TCP / IP and Ethernet Networking", Power Engineering Journal, vol. 16, no. 6, pp. 305-311, 2002.

[5] S. Chen, F.Y. Lu, "Web-Based Simulations of Power Systems," IEEE Trans. Computer Application in Power, vol. 15, no. 1, pp. 35-40, 2002.

[6] Keinosuke Matsumoto, Tomoaki Maruo, Naoki Mori, Masashi Kitayama and Yoshio Izui, "A Communication Network Model of Electric Power Trading Systems using Web Services," IEEE power Tech conf. Proceedings, vol. 3, pp. 1-6, 2003.

[7] Quirino Morante, Alfredo Vaccaro, Domenico Villacci and Eugenio Zimeo, "A Web based Computational Architecture for Power Systems Analysis," Bulk Power System Dynamics and Control - VI, August 22-27, 2004.

[8] S. Chen, "Open Design of Networked Power Quality Monitoring Systems," IEEE Trans. Instrumentation and Measurement, vol. 53, no. 2, pp. 597-601, 2004.

[9] K. Nithiyanandan and V. Ramachandran, "RMI Based Multi-Area Power System Load Flow Monitoring," Iranian J. Electrical and Computer Engineering, vol. 3, no. 1, pp. 28-30, 2004.

[10] Kannan Nithiyanandan and Velimuthu Ramachandran, "RMI Based Distributed Model for Multi–Area Power System On–Line Economic Load Dispatch," J. Electrical Engineering, vol. 56, no. 1-2, pp. 41–44, 2005.

[11] Afaneen A. Abood, Ahmed N. Abdalla and Shant K. Avakian, "The Application of Multi-Agent Technology on Transient Stability Assessment of Iraqi Super Grid Network", American J. Applied Sciences, vol. 5, no. 11, pp 1494-1498, 2008.

[12] M. Shekhar, S.S.K. Kelapure, Sastry Akellay and J. Gopala Raoz, "Application of Web Services in SCADA Systems," Int. J. Emerging Electric Power Systems, vol. 6, no. 1, pp 1-15, 2006.

[13] Jun Zhai, Jianfeng Li, Qinglian Wang, "Using Ontology and XML for Semantic Integration of Electricity Information Systems," 3rd Int. Conf. Electric Utility Deregulation and Restructuring and Power Technologies, pp. 2197-2201, 2008.

[14] F. Milano, L. Vanfretti, "Open Model for Exchanging Power System Data", presented in IEEE Power & Energy Society General Meeting, 2009

[15] P. Kundur, J. Paserba, V. Ajjarpu, G. Anderson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Tankovie, C. Taylor, T.V. Cutsem and V. Vittal, "Definition and Classification of Power System Stability," IEEE Trans. Power System, vol. 19, no. 3, pp. 1387-1401, 2004.