Transmission system data (pu):

$ZL1 = 0.0333 + j0.5 \quad ZL2 = 0.0620 + j0.93 \quad Ztr = 0.01 + j0.15.$

Prefault system condition (pu):

$E_t = 1.0 \quad P = 0.9 \quad Q = 0.3631(\text{overexcited}) \quad Eb = 0.90081.$

Constants of the vector control:

$$K_\delta = 1.1 \quad K_{Pm}^{p} = 1.1.$$

Figure 3 illustrates the performance for a stable case, showing that the rotor angle oscillations can be damped significantly due to a vector control action being activated at less than 0.5 seconds. In addition, an unstable case, which becomes stable after a vector control action of about 0.5 s, is presented in Figure 4.

Conclusions: The application of the vector control approach to reduce synchronous generator oscillations can be effective and constitutes a promising alternative to present methods used to enhance the transient stability of power systems. Although the control strategy was developed and tested considering a SMIB system, the approach can be extended to multimachine systems with appropriate modifications.

References:

[1] P. Kundur, *Power System Stability and Control.* New York: McGraw-Hill, 1994.

[2] D. W. Novotny and T. A. Lipo, *Vector Control and Dynamics of AC Drives.* Oxford, UK: Oxford Univ. Press, 1997.

 Manuscript received 1 September 1999. This paper is published herein in its entirety.

XML and Data Exchange

E.Z. Zhou

Author Affiliation: Norrell Information Services, Troy, MI.

Abstract: The world has entered the Web, Internet era. In the near future power system analysis and simulation could be performed in distributed fashion over the internet. To realize this vision, the first step has to be to define open extensible standards for data exchange. How to represent power system simulation data and results for data exchange between different simulation programs and/or different companies has been a subject of interest for many years. So far, efforts to define "common" formats for the exchange have had limited success, due to lack of extensibility. Recently, there has been a lot of excitement about the extensible markup language (XML), which among other things, has the ability to define new tags for representing business problems. This letter will explore the idea of using XML as a base for data exchange for power system analysis and simulation.

Keywords: XML, data exchange, load flow.

Introduction: The world has become a small village with regard to information exchange, thanks to the explosive use of the Web and Internet. It is possible that, in the not too-distance future, some one could sit in his/her office anywhere in the world with an Internet connection, access databases, and use analysis modules physically located and served on different servers at different locations to perform his/her daily analysis and simulation work. To realize this vision, the very first step has to be to define open extensible standards to exchange data.

How to represent power system simulation data and results for data exchange between different simulation programs and/or different companies has been a subject of interest and concern for many years. For example, although basic information required for load flow calculation has remained unchanged for last 30 years, previous efforts on defining "common" formats for load flow data exchange, such as IEEE common format, have had limited success. This author believes that the main reason for the limited success is due to lack of extensibility. Recently, there has been a lot of excitement about the extensible markup language (XML), which among other things, has the ability to define new tags, or even define a whole new markup language, for a particular set of business problems. The idea of using XML as a base for data exchange for power system simulation will be explored in this letter. A load flow example will be used to outline the key concepts.

Extensible Markup Language (XML): Most of us are familiar with markup tags from HTML. For example, web page authors often use tables to present their information. The following is an HTML source code example.

```
<Table Border=1>
  <Tr>
    <Th>Name</Th> <Th>Wage</Th> <Th>Sta-
tus</Th>
  </Tr>
  <Tr>
    <Td>Mike Zhou</Td> <Td>$6.00/Hr</Td>
<Td>Part-time</Td>
  </Tr>
</Table>
```

HTML Code Listing-1

A browser may render the above HTML code as follows:

Name	Wage	Status
Mike Zhou	$6.00/Hr	Part-time

XML is a text-based format for structured documents. It is also a markup language; it uses markup tags as well. But unlike HTML, it allows content authors to define their own tags. XML tags describe the content, rather than the presentation of that content.

0272-1724/00/$10.00©2000 IEEE

XML Element: An XML element is made up of a start tag, an end tag, and data in between. The start and end tags describe the data within the tags. The data within the tags is considered the value of the element. For example, the XML element below is a "PoorGuy" element with the value of "Mike Zhou":

```
<PoorGuy>Mike Zhou</PoorGuy>
```

The element name "PoorGuy" allows you to mark up the value "Mike Zhou" semantically, so that you can differentiate that particular bit of data from another, similar bit of data.

XML Document: A basic XML document is simply an XML element that may or may not include nested XML elements. For example, the XML element "Person" below is a XML document, which contains the same information as HTML Code Listing-1, but without any indication of how it should be presented:

```
<Person>
  <Name>Mike Zhou</Name>
  <Wage length="hour" unit="$">6.00</Wage>
  <Status>Parttime</Status>
</Person>
```

XML Code Listing-2

Well-Formed and Valid XML: Figure 1 shows a simple power network. The following is a well-formed XML file for describing the network configuration and data.

```
<?xml version="1.0"?>
<Network>
    <BusList>
     <Bus id="Bus1">
      <BusData> ... </BusDate> </Bus>
     <Bus id="Bus2">
      <BusData> ... </BusDate> </Bus>
    </BusList>
    <BranchList>
     <Branch fromid="Bus1" toid="Bus2">
      <BranchData> ... </BranchData> </Branch>
    </BranchList>
</Network>
```

XML Code Listing-3

An XML document can be both well formed and valid. To be well formed, the XML needs simply adhere to the syntax rules as laid out in the XML specification. To be valid for certain purpose, an XML document must adhere to some logical structure, which could be described by a document type definition (DTD) file. The corresponding DTD file for describing the simple network is as follows:

```
<!ELEMENT Network(BusList+, BranchList+)>
    <!ELEMENT BusList(Bus+)>
     <!ELEMENT Bus (BusData)>
     <!ATTLIST id ID #REQUIRED>
      <!ELEMENT BusData>
    <!ELEMENT BranchList(Branch+)>
     <!ELEMENT Branch (BranchData)>
     <!ATTLIST fromid #REQUIRED toid #REQUIRED>
      <!ELEMENT BranchData>
```

DTD Code Listing-4

The DTD syntax is not so straightforward and needs some explanation. The DTD states that a Network element must include one or more BusList element(s), and one or more BranchList element(s). Each BusList element must include one or more Bus element(s). Each Bus element must have a unique id attribute and a BusData element. A Branch element must have a fromid attribute, a toid attribute, and a BranchData element. The DTD gives much more control over the format/structure of an XML document for certain purpose. The DTD can be used to check the validity of an XML document.

DOM XML Parser: XML is a structured markup language. The markup in an XML file has a particular structure that means something to the applications that process it. The structure of XML is so simple and consistent that it's possible to represent any XML document as a tree of objects in any object-oriented programming language. The World Wide Web Consortium (W3C) has defined a complete set of objects to be used for processing XML documents. The specification for this set of objects is called the document object model (DOM).

In general, you can think of an XML document as a tree structure, with the "contains" relationship in the document corresponding to a parent-child relationship in the tree. Figure 2 shows the tree structure of the network information enclosed in the XML Code Listing-3.

Any well-formed XML document can be represented in terms of DOM as a tree object in memory by using a DOM XML parser. Then the document could be processed as a tree of nodes, instead of as a series of lines or tokens. Fortunately, we don't have to go out and write a parser to convert an XML file to a DOM object from scratch. Several companies and individuals have written parsers that read an XML document from a file and produce a DOM Document object. IBM's java XML parser, which is available free for noncommercial use from IBM's alphaWorks web site, for example, implements the W3C DOM specification completely and comes with excellent documentation.

A Load Flow Data Exchange Example: Load flow calculation is the foundation for power system simulation. Figure 3 shows a simple power network for load flow calculation. The following is an XML document for describing the network configuration and data.

```
<?xml version="1.0"?>
<LFNetwork id="5-Bus LF Test System">
    <BaseKVA value="100" unit="mva"/>
    <BusList defaultBaseV="13800" unit="volt">
                <Bus id="1" type="pq">
```

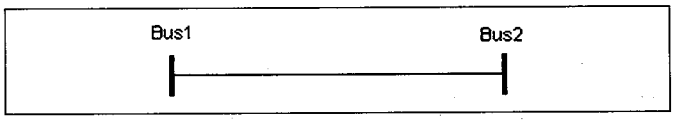

Figure 1. A simple network

2000 Control Center Workshop

8-10 May, Scottsdale, Arizona

The sixteenth biennial Control Center Workshop will be held in Scottsdale, Arizona, on 8-10 May 2000. Anyone who is interested in or involved with control centers of any type should attend this meeting. The theme of the workshop is: "Life in the New Millennium: How to Survive in the ISO/RTO Environment." The workshop is sponsored by the Energy Control Center Subcommittee of the PES Power System Operations Committee, Salt River Project, Arizona State University, and Power Computing Applications, Inc.

There will be three sessions to the Workshop. Session 1 will be chaired by Ralph Masiello of ABB and will provide an overview and update on the ISO and deregulation activities in North America. Session 2 will be chaired by Steve Widergren of Alstom ESCA & Joe Bucciero of KEMA Consulting and will cover the new technologies and standards involved in surviving in this environment. Session 3 will be chaired by Joann Staron of PCA and will cover the new algorithms and programs needed to make a go of it in today's world.

For more information, contact the technical program chair, Ed Dobrowolski of KEMA Consulting, E-mail edobrowolski@kemaconsulting.com.

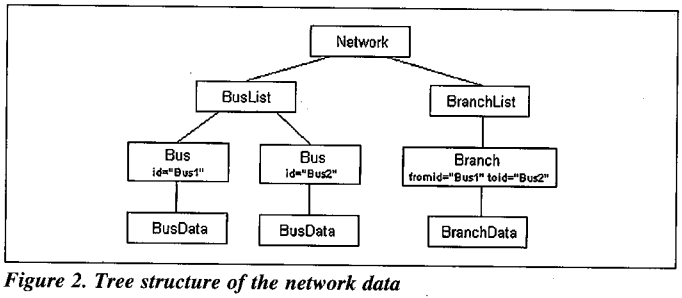

Figure 2. Tree structure of the network data

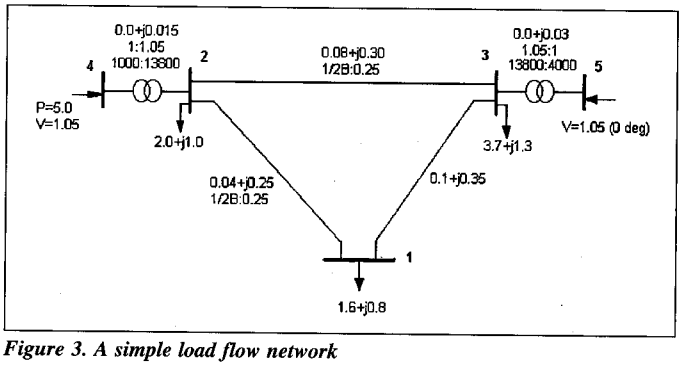

Figure 3. A simple load flow network

```
      <P value="1.6" unit="pu"/>
      <Q value="0.8" unit="pu"/> </Bus>
     ...
     <Bus id="4" type="pv">
      <BaseV value="1000" unit="volt"/>
      <VMag value="1.05" unit="pu"/>
     <P value="5.0" unit="pu"/> </Bus>
      <Bus id="5" type="swing">
       <BaseV value="4000" unit="volt"/>
       <VMag value="1.05" unit="pu"/>
       <VAng value="0.0" unit="deg"/> </Bus>
    </BusList>
    <BranchList>
     <Branch fromid="1" toid="2" type="line">
      <R unit="pu" value="0.04"/>
      <X unit="pu" value="0.25"/>
       <HalfBShunt  unit="pu"  value="0.25"/>
</Branch>
      ...
     <Branch fromid="5" toid="3" type="xformer">
      <X unit="pu" value="0.03"/>
       <ToTapRatio  unit="pu"  value="1.05"/>
</Branch>
   </BranchList>
  </LFNetwork>
```

XML Code Listing-5

Tags that end with a "/>" are empty tags. That is, there's nothing that can be placed between an opening and closing tag of that type. The following conceptual steps may be used to construct a load flow calculation process:

Step-1: Input a load flow data XML file;
Step-2: Use an XML parser to build a DOM tree object;
Step-3: Extract load flow data from the DOM object;
Step-4: Run load flow calculation;
Step-5: Write load flow results back to the DOM object;
Step-6: Generate a load flow results XML file from the DOM object.

```
<?xml version="1.0"?>
<LFNetwork id="5-Bus LF Test System">
    ...
     <Bus id="1" type="pq">
      <VMag value="0.8622" unit="pu"/>
      <VAng value="-4.78" unit="deg"/>
      <P value="1.6" unit="pu"/>
      <Q value="0.8" unit="pu"/>
     </Bus>
    ...
</LFNetwork>
```

XML Code Listing-6

Load flow result XML file may look like the above. It demonstrates how to extend or the extensibility of XML. One can add properly marked-up information without any limitation to extend an XML file. For load flow input, a pq-bus only needs <P> and <Q> tags; for load flow results, <VMag> and <VAng> tags need to be added. For example, for one's own company use, some extra information needs to be added to a bus. This could be accomplished by adding a new XML element as follows:

```
<?xml version="1.0"?>
<LFNetwork id="5-Bus LF Test System">
    ...
     <Bus id="1" type="pq">
      <VMag value="0.8622" unit="pu"/>
      ...
      <MyCompInfo>...</MyCompInfo>
     </Bus>
    ...
  </LFNetwork>
```

XML Code Listing-7

Summary: XML enables the separation of the meaning or semantics of the data from the way it is used by an application or rendered for presentation. One can use XML tags to define what one's data means by using the natural vocabulary of one's business problem domain, for example, using power system simulation terms. This is the key motivation for XML's invention and the basis of its usefulness. From the business perspective, almost any type of data can be represented as XML, with a grammar (DTD) to describe its structure. The load flow example clearly outlined the flexibility and extensibility of XML. This author believes that flexibility and extensibility are crucial factors for the success of any standard for data exchange. A recent report suggests that XML will revolutionize the exchange of business information similar to the way the phone, fax machine, and photocopier did when those devices were invented. Those prior inventions made a significant impact on how business viewed and exchanged information. XML is poised to impact the Internet area the same way.

 Manuscript received 29 June 1999. This paper is published herein in its entirety.

Guidelines for a Good Poster/Paper Session

Helpful guidelines for running a good oral or poster session and for making a good presentation (including slide preparation) are available on the Web. Start at the PES home page, http://www.ieee.org/power, then go to: PES Author's Kit. At the bottom of this page find, as necessary: Guide for Presenter, Guide for Slides, or Guide for Session Chair.

The guidelines should help the chair in running a smooth session and the presenter in effectively conveying the desired message to the audience.