

Import and Export PSCAD Document Based on XML Technology

Weimei Zou^{1, a}, Min Liu^{1, b}, Lixiang Sun^{1, c}, Dongyue Yang^{1, d}, Guochao Xie^{2, e},
Jiayu Wang^{2, f} and Chongru Liu^{2, g}

¹China Electric Power Research Institute, Beijing 100192, China;

²School of electrical and electronic engineering, North China Electric Power University, Beijing 102206, China.

^azouwm@epri.sgcc.com.cn, ^bliumin@epri.sgcc.com.cn, ^csunlixiang@epri.sgcc.com.cn,
^dyang_dongyue@163.com, ^eguochaoxie@163.com, ^fsuewjy@163.com,
^gchongru.liu@ncepu.edu.cn

Abstract. To figure out a way to solve the problem that the model of a system built in a simulation software is needed to simulation in another software, a transfer technique based on XML is proposed in this paper. It can help the users transferring simulation system from PSCAD between other simulation software automatically and conveniently. The import and export of XML data as well as its model transformation are the two parts of the transfer technique. This paper mainly describes the XML data import and export. It is implemented by using libxml2 function library of the Microsoft visual studio platform. First, study the structure and content of the PSCAD document; second, use libxml2 function library to import PSCAD file data; third, export PSCAD file data in the DOM parsing mode. Simulation results show that the import and export process are both correct.

Keywords: Electromagnetic transient; Libxml2; PSCAD; Simulation; XML.

1. Introduction

PSCAD/EMTDC (Power Systems Computer Aided Design) is power system electromagnetic transient simulation software which is widely used in the world. EMTDC (Electromagnetic Transients including DC) is the core of simulation calculation and PSCAD provides graphical user interface to EMTDC [1]. Not only does its visual interface supply power system electromagnetic transient simulation with good operating interface, and its programmable and customizable function offers abundant and flexible model selection for the research of power system electromagnetic transient process. EMTDC uses the FORTRAN compiler, which takes the advantage of the compiler so that generated program is very suitable for numerical calculation.

The advantage of electromagnetic transient (EMT) simulation is the high accuracy. PSCAD is one of the widely used EMT simulation software. However, the calculation time of EMT simulation software are too long for large scale power system and the electromechanical simulation tool is lack of accuracy for some electronic devices. Therefore, the hybrid electromechanical electromagnetic transient simulation software becomes an alternative choice for large scale power system [2, 3]. Since PSCAD has been used for a long time, a lot of large scale systems are of PSCAD data form. Therefore, it is necessary to transfer both data and graph automatically between PSCAD and hybrid electromechanical electromagnetic transient simulation software, which is very inconvenient now.

Considering that PSCAD X4 uses the XML technology to organize data file [4], the import and export PSCAD document method based on XML technology is presented in this paper. The remainder of this paper is arranged as follows, section 2 gives a brief introduction to the structure and content of the PSCAD file and the libxml2 function library. Section 3 indicates the detail process of the data transformation. Section 4 shows the simulation results by comparing the dynamic behaviors and the topology before and after conversion and section 5 draws the conclusions of this paper.

2. Introduction of Structure and Content of PSCAD File and Libxml2 Function Library

2.1 Structure and Content of the PSCAD File.

Since X4 version, PSCAD has used the new XML-based file format to store data. XML is a widely used extensible markup language and it encodes data into plain text format [4]. The structure of PSCAD file is a tree structure, the root element of which is "project". In addition, there are three essential child elements, "paramlist" which uses to record running information, "hierarchy" which uses to record hierarchy information, and "definitions" which uses to record component information. Each hierarchy corresponds to each "Definition", which is child element of "definitions". The content is stored in the attributes with the form of name-value pairs. A simplified XML tree of PSCAD case including a resistor is presented below.

```
<project Target="EMTDC" name="example">
    <paramlist name="Settings">
        <param name="time_duration" value="0.5"/>
    </paramlist>
    <hierarchy>
        <call name="Station" link="728011497">
            </call>
    </hierarchy>
    <definitions>
        +<Definition name="Station" view="false" instances="1" classid="StationDefn">
        <Definition name="Main" view="false" instances="1" classid="UserCmpDefn" date="0">
            <schematic classid="UserCanvas">
                <User name="master:resistor" id="1804014950" defn="master:resistor" orient="0"
                    h="29" w="52" y="144" x="306">
                    <paramlist name="" link="-1">
                        <param name="R" value="1.0 [ohm]"/>
                    </paramlist>
                </User>
            </schematic>
        </Definition>
    </definitions>
</project>
```

Fig. 1 XML tree of PSCAD

2.2 Libxml2 Function Library.

There are two main popular XML file parsers to export the data from an xml file effectively, DOM and SAX [5]. DOM is an object model that is used to access any location in the XML file. In the DOM, XML data exists with a tree structure, where XML file can be viewed that it consists of XML nodes and its data can be stored in the content or properties. DOM parsing mode builds a complete parse tree in memory to achieve a comprehensive and dynamic access to entire XML document. However, with the increasing size of XML documents, it will lead to memory consumption and increase processing time [6, 7]. SAX is an event-driven API. SAX works in a similar method with streaming. Analysis can start at any time and does not need to wait until all the data files processed. Since the SAX parsing mode is sequential, SAX provides a mechanism for sequential access, which cannot go back and reprocess for already parsed part, so that SAX parsing mode is simple but inflexible. Thus, this paper chooses DOM parsing mode to carry out research [8, 9].

Appropriate XML parsing technique has greater impact on the overall performance of the parsing process, while inappropriate option could consume too much running time. On the basis of depth first searching on parsing principles and programming mechanism of different XML parser, the libxml2 function library is utilized to automatically parse PSCAD data of XML format between ADPSS data which also organizes data in XML format [10].

Libxml2 function library is a free and widely used C language parser to process XML, and it is open source software. It can be used for C and C++ environment, and can run on Windows platform [11]. A number of functions is packaged in Libxml2 function library. By traversing the DOM tree to realize functions, it is able to meet the requirements of most developers [12].

Some specific data types of libxml2 list below:

- 1) `xmlChar` substitutes `char`;
- 2) `xmlDoc` is a structure tree creating after parsing a file, and `xmlDocPtr` is a pointer to the structure tree;
- 3) `xmlNode` contains the structure of a single node, and `xmlNodePtr` is its pointer.

The main functions of libxml2 list below in Table.1:

Table 1 Main functions of libxml2

Name	Function
<code>xmlDocPtr xmlDocParseFile(const char * pdoc)</code>	Store the XML file in memory in the form of the structure of the DOM tree.
<code>xmlNodePtr xmlDocGetRootElement(xmlDocPtr doc)</code>	Obtain the root element.
<code>xmlChar* xmlGetProp(xmlNodePtr node, const xmlChar * name)</code>	Obtain the attributes of the element.
<code>xmlAttrPtr xmlSetProp(xmlNodePtr node, xmlChar* name, xmlChar* value)</code>	Set the attribute of the nodes; if the attribute already exists, replace its value.

3. Import and Export Process

3.1 Importing PSCAD File.

As explained in the previous section, obtaining running information is relatively easy, which only needs to traverse the element name and read attributes, and then assign the two values to the defined variables.

When obtaining definition information, first, identify which hierarchy the element belongs to by recognizing the attribute name of element “Definition”, and then traverse whole components of every “Definition” by identifying component name, finally, read the component parameters and assign the values to variables defined in the head file. The code of reading resistor component is presented below.

```

xmlNodePtr curN1,curN2;
while(curN1!=NULL)
{
    if(!(xmlstrcmp(curN1->name, "paramlist")))
    {
        curN2=curN1->xmlChildrenNode;
        while(curN2!=NULL)
        {
            if(!(xmlstrcmp(curN2->name,
"param")))
            {
                get_name_value(curN2,data);
            }
            curN2=curN2->next;
        }
        curN1=curN1->next;
    }
    map<string,string>::iterator iter;
    iter = data.find("R");

```

Fig. 2 Code of reading resistor component

When obtaining hierarchy information, the most important issue is to determine the affiliation between the hierarchies. If current hierarchy has child hierarchy, the level of child hierarchy is one more than the level of current hierarchy. At the same time, current hierarchy is recorded as the parent hierarchy of child hierarchy. If current hierarchy has brother hierarchy, the level of brother hierarchy

is equal to the level of current hierarchy and these two hierarchies have same parent hierarchy. It reads the hierarchy information line by line with recursive function, and records the required information. For each hierarchy the recursive function is called to determine whether there are child hierarchy and brother hierarchy. The flowchart of the process of obtaining hierarchy information is shown below in Fig.3:

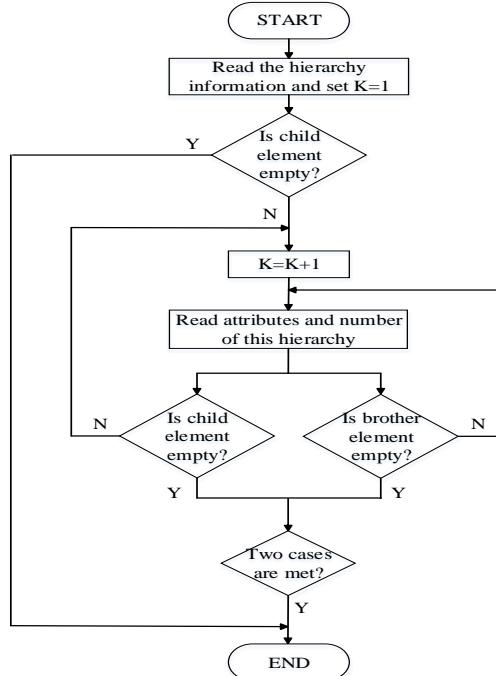


Fig. 3 Program flowchart of reading hierarchy

It can be seen that the variable K records the level of hierarchy and helps to determine whether the child hierarchy and brother hierarchy exist or not.

3.2 Exporting PSCAD File.

Writing PSCAD file has two ways. According to the format PSCAD required, one way is calling libxml2 library functions to write data line by line, the other is first reading PSCAD empty file that has no component, then modifying the default data of empty file for the target data, finally writing hierarchy information and components information to the file. Taking the basic format and default data of PSCAD empty file into account, the latter one is chosen. First, use libxml2 function to load the PSCAD empty file and modify the default running information and hierarchy information with target data. Then according to hierarchy information, create element "Definition" and write stored component element to corresponding "Definition". It writes the element "hierarchy" one by one. By identifying the level of hierarchy, it determines the affiliation between the hierarchies and puts current hierarchy into correct position. The flowchart of the process of writing hierarchy information is shown below in Fig.4:

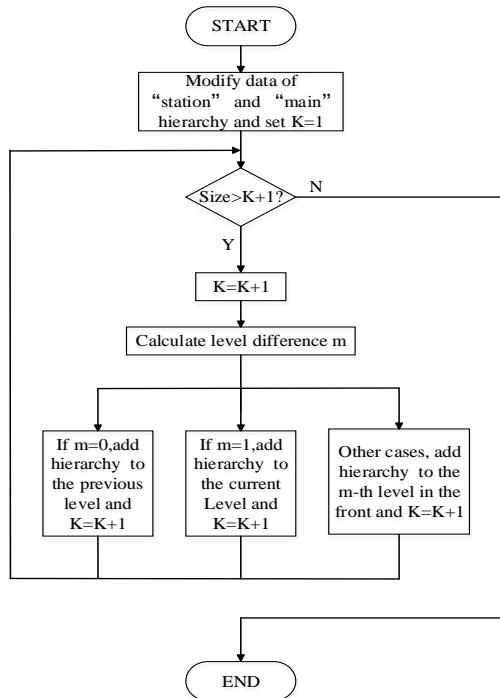


Fig. 4 Program flowchart of writing hierarchy

The variable *size* represents the number of hierarchy and the variable *K* represents the order of reading hierarchy information. By calculating the level difference *n*, it selects the writing position of new target hierarchy.

4. Simulation Results

PSCAD file has many hierarchies. Each hierarchy has a variety of components. Wires connecting to the different components can be numerous and signals can be transferred among different hierarchies.

The method to store the PSCAD data is depicted as follows, for each kind of component, it designs a structure, records not only physical parameters, but also affiliation of hierarchies, direction and coordinates of each component as well as number of the component; For each hierarchy, it designs a structure, defines variables to record basic information of each hierarchy, and defines vectors to point at component structures. According to this method the components and topological relations of original file can be stored in RAM.

Cigre_Benchmark system is adopted to compare simulation results before and after conversion. It has 29 running information, 7 hierarchy information and 7 definition information. This model uses unipolar 500kV, 1000MW HVDC, rectifier and inverter side has adopted a 12-pulse converter, connected to weak AC system. This system is widely applied to test the current system. First, import PSCAD file with libxml2 library and compare the level of hierarchy and number of components of each hierarchy in computer RAM with original PSCAD file. Then export the PSCAD data in RAM to document and test the dynamic behaviors. Fig. 5 depicts main circuit diagrams before conversion.

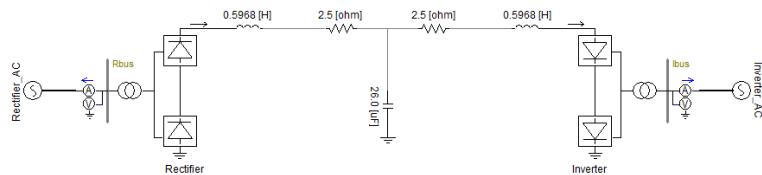


Fig. 5 Topology of Cigre_Benchmark system before conversion

After importing PSCAD file, wire structure in RAM records coordinates of its starting point and ending point. The number of wire structure represents how many wires in original file. The center coordinate and direction of the component can determine the pin coordinates of component. If the two pin coordinates of different components are consistent with the coordinates of starting point and

ending point of a wire, the two components are connected with the wire. Thus, the topology and components of each hierarchy can be stored correctly in RAM.

Signals among the hierarchies can be transferred with the wireless connection. If two hierarchies are needed to transfer a signal, then it sets the signal as wireless connection and puts this signal into two hierarchies. After importing PSCAD file, two hierarchies have the structure of wireless connection of this signal in RAM.

Table.2 shows the level of hierarchy and number of components and wires of each hierarchy in computer RAM, which is identical to the original PSCAD document.

Table. 1 Detail information in computer RAM

Hierarchy No.	Hierarchy	Level	Components number	Wires number
1	Station	1	0	0
2	Main	2	8	4
3	Inverter_AC	3	9	6
4	Inverter	3	32	26
5	InvCtl	4	44	41
6	Rectifier	3	49	26
7	Rectifier_AC	3	6	6

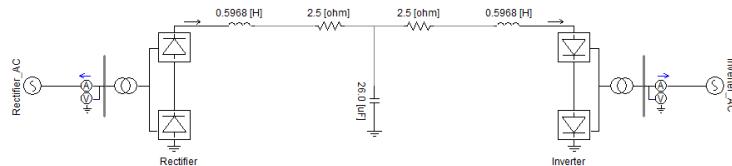


Fig. 6 Topology of Cigre_Benchmark system after conversion

The topology is compared in figures respectively. It can be seen that all components after conversion have same positions in original PSCAD file. Therefore, the topology conversion is correct by the XML parsing based on libxml2 library.

This paper focuses on the rectifier DC and AC voltage and current waveforms. Fig. 7 and Fig. 8 depict the comparison of the rectifier DC voltage and current waveform before and after conversion. In addition, the comparison of the rectifier AC voltage waveform is indicated in Fig. 9. The simulation time is 10s. The dynamic behaviors of simulation after conversion are consistent with those before conversion. Therefore, the simulation results show the proposed method is of feasibility practice engineering.

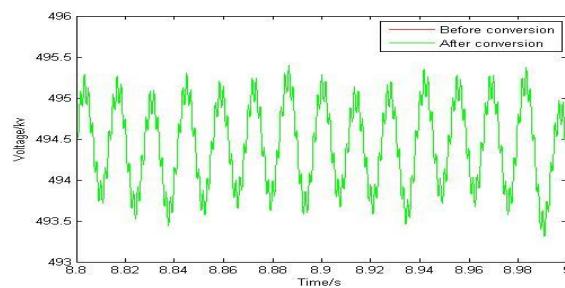


Fig. 7 Rectifier DC voltage waveform comparison

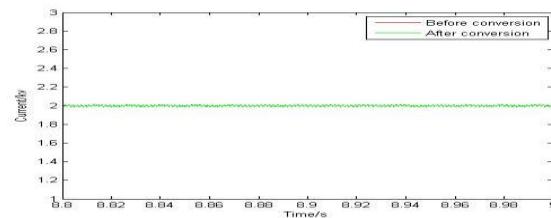


Fig. 8 Rectifier DC current waveform comparison

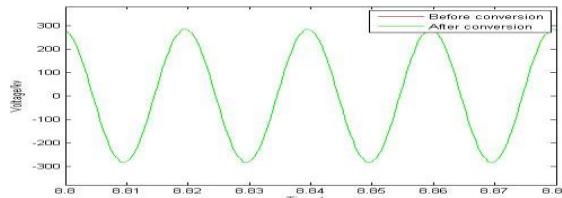


Fig. 9 Rectifier AC voltage waveform comparison

5. Conclusion

This paper focuses on a PSCAD file parsing technique based on XML. Because XML documents can be marked the properties of the document, more importantly, it marks the structure and character of the document, thus it is convenient to operate, modify and maintain. Compared with the way of parsing PSCAD file character by character, parsing based on XML makes the program simple and clear and improves efficiency and accuracy.

Libxml2 library supports two common analytical methods. Although DOM parsing mode needs high memory, and the complex structure of the tree takes traversal time, because the structure of tree used in DOM analytical method is similar to the structure of XML document tree, DOM parsing mode has been widely used. Additionally, the random access mechanism supplies great flexibility to program development.

References

- [1] LIN Liangzhen, YE Lin. Electromagnetic transient analysis software package PSCAD/EMTDC [J]. Power System Technology, vol. 24, pp. 65-66, Jan. 2000.
- [2] TAO Hua, XU Jinjin, ZOU Wencong. Model conversion from BPA to PSCAD [J]. Electric Power Automation Equipment, vol. 8, pp. 152-156, 2013.
- [3] LIU Yongjun, LIANG Xu, Min Yong, and et al. Power system electromechanical transient and electromagnetic transient hybrid simulation interface algorithm [J]. Automation of Electric Power System, vol. 30, pp. 44-48, Jun. 2006.
- [4] KONG Lingbo, TANG Shiwei, YANG Dongqing, and et al. XML data query technology [J]. Journal of Software, vol. 18, no. 6, pp. 1400-1418, Jun. 2007.
- [5] MENG Xiaofeng, WANG Yu, WANG Xiaofeng. XML Query Optimization [J]. Journal of Software, vol. 17, no. 10, pp. 2069-2086, Oct. 2006.
- [6] HOU Shuang, WANG Lancheng, and TIAN Mei, and et al. XML schema database advantage and XML document conversion [J]. Journal of Shanghai Jiaotong University, vol. 37, pp. 195-197, Sep. 2003.
- [7] S. Dennetiere, A. Parisot, E. Milin. "Use of XML to import SCADA data into EMTP," 2011 IEEE Power & Energy Society General Meeting. IEEE, pp. 1-5.
- [8] LI Cong. "The design and implementation of XML-based data exchange platform," Master thesis, Wuhan University of Technology, 2009.
- [9] WANG Ruijie. "Analysis and Implementation of XML data conversion technology," Master thesis, University of Electronic Science and Technology, 2010.
- [10] TAO Hua, YANG Zhen, ZHANG Min, and et al. Implementation of power system spanning tree based on depth-first search algorithm [J]. Power System Technology, vol. 34, pp. 120-124, Feb. 2010.
- [11] CHEN Hongmei, CHEN Jing. XML document parsing technology [J]. Information technology, vol. 10, pp. 175-177, 2009.
- [12] ZHANG Miao, HUI Xiaoqiang. A fast algorithm for XML document validation [J]. Computer Technology and Development, vol. 8, pp. 123-127, Aug. 2015.