

# Open Model For Exchanging Power System Data

F. Milano, *Member, IEEE*, M. Zhou, *Member, IEEE*, and GuanJi Hou

**Abstract**—This paper presents an XML-based open data model for exchanging power system simulation data. The need for an open model born due to both the limitations of the IEEE common data format and the huge number of proprietary formats that make difficult exchanging system data in the power system community. The philosophy, the structure and the features of the proposed open data model are discussed in detail. The proposed open data model is illustrated by means of a small-size benchmark network.

**Index Terms**—Data format, XML schema, power systems simulation.

## I. INTRODUCTION

### A. Motivation

IN the early seventies, the IEEE recommended a Common Data Format (IEEE CDF) for exchanging power flow data [1]. Unfortunately, the IEEE CDF is not flexible and is limited to static data. These facts make this format not suitable for storing dynamic and market information. Besides, software companies use their own internal data format. Most of these formats are proprietary and generally are not freely documented. We think that the power system community needs a modern, completely free, flexible and well-documented format for power system analysis data exchange. The extensible markup language (XML) is a natural choice since it has reached a very mature technology and has become a worldwide standard for exchanging and sharing information. Furthermore, a huge variety of open-source XML processing tools are freely available. Surprisingly enough, the power system community is one of the very few scientific and technical areas where XML is not widely used. For those who are unfamiliar with XML, some introductory information can be found in [2], while a discussion on the usage of XML for power systems is [3].

### B. Literature Review

Several formats for power system data have been defined. Typically, companies that provide commercial software packages for power system analysis have their own proprietary data format. Examples of proprietary formats are PSS/E [4], ABB Simpow [5], DigSilent [6], Neplan [7], GE-PSLF [8], CYME [9], Eurostag [10], PowerWorld [11], CEPEL [12], PST [13], [14], etc. Also some system operators that have created their

own software analysis tools, have defined their own data format. Some examples are ENEL in Italy (the INPTC1 format [15]), and the recent European Union for the Coordination of Transmission of Electricity (UCTE format) [16]. Also research-oriented software tools have defined their own formats; for example PSAT [17], MatPower [18] and the Power Education Toolbox (PET) [19]. This Babel of data formats have been originated mostly because of the commercial nature of most of the software packages for power system analysis. Commercial software packages seldom allow to import or to export data in other formats. One exception is the PSS/E format that can be imported by several applications, since PSS/E is often recognized as a standard *de facto*. However, there has not been a real incentive to define a standard for power system analysis, except for the IEEE common data format (CDF) [1]. This format dates back to the 70s and unfortunately is limited to only static power flow data.

In this paper, we show that defining a common data scheme is a preferable and more efficient approach than the current methods for exchanging data files. We also briefly describe the advantages and the drawbacks of several existing data formats and show that the proposed data scheme keeps all the advantages and is able to avoid the drawbacks of other formats.

### C. Contributions

In summary, these are the novel contributions of the paper.

- 1) The proposal of an open and extensible data model based on XML for exchanging power system data information.
- 2) A robust structure that will be used to build and extend the proposed data model.
- 3) An example of the current implementation of the ODM for power flow.
- 4) To provide a reference document to the power system community about the existence of the ODM.

### D. Paper Organization

This paper is organized as follows. Section II describes common concepts of data formats and the proposed idea for defining an open data model. Section III presents definitions and common concepts used in the XML schema for the proposed ODM. In Section IV, the proposed model is illustrated through a simple network and its features are discussed in detail. Section V presents a variety of adapters that convert data to and from the ODM. Finally, in Section VI, conclusions are duly drawn.

## II. BACKGROUND

In this section, we discuss the state of art of the formats used for power system analysis, the features and the advantages of the proposed ODM schema.

Submitted to 2009 IEEE PES General Meeting. This paper is part of the activity of the IEEE Task Force on Open Source Software.

F. Milano is partly supported by the Ministry of Science and Education of Spain through CICYT Project DPI2006-08001; and by Junta de Comunidades de Castilla-La Mancha, through project PCI-08-0102.

M. Zhou is with TIBCO Software, Inc., Palo Alto, CA, USA (e-mail: mike@interps.org).

GuanJi Hou is with the College of Electrical Engineering, South China University of Technology, Guangdong, China (e-mail: stephen@interps.org).

## A. State of the Art of Data Formats for Power System Analysis

Among all existing formats, one can identify a few basic groups, as follows. The taxonomy can be made based on different features. The most important features are described in the list below. Table I provides a synoptic scheme of the features of a variety of formats for power system analysis.

### 1) The way data are stored, organized and structured:

- *Fixed position, fixed order.* Any data has its starting and ending columns and is assigned a maximum number of digits. Also the order in which data are listed is fixed. For example, in the IEEE CDF the bus card comes before the branch card, and the voltage magnitude must occupy columns 28 to 33 of the bus card [1]. The formats created by WSCC-EPRI [20], Eurostag [10], UCTE [16], and ENEL (INPTC1) [15] are other examples of this kind of format.
- *Free position, fixed order.* The position of the data is free, however, the order in which the data is listed in the file is fixed. For example, in the PSS/E format, the load data must follow the bus data, and the generator data must follow the load data [4]. In the bus data, the bus name must follow the bus identification number, etc. However, there is no restriction on the token position, and data can be separated by commas or by spaces. Other examples of this kind of formats are those of GE-PSLF [8] and FlowDemo.net [21].
- *Free position, free order.* The position and the order of the data are free. Typically, an *ad-hoc* parser is needed to read this kind of data files, which makes quite difficult to create an import/export utility. Some examples of such formats are the formats used by Simpow [5], DigSilent [6], PowerWorld [22] and InterPSS [23]. A special example of this format are also all data files written in Matlab (see for example, PSAT [17], PST [24], and MatPower [18]).

### 2) The kind of data and analysis supported:

- *Static Data.* (e.g. power flow data).
- *Dynamic Data.* (e.g. synchronous machine and regulator parameters).
- *Market Data.* (e.g. generator and load bids).
- *Short Circuit Analysis Data.* (e.g. negative and zero sequence of generators and transformers).
- *Graphical Data.* (e.g. network scheme, geographical information system, etc.).
- *Other Data.* (e.g. FACTS data, user defined component data, etc.).

### 3) The number of files that compose the full system data set:

- *Single file.* Most of the data formats requires a single file for defining the whole network. This is typical of most formats.
- *Multiple fixed number of files.* Some formats use different files for different information. For example the power flow data is in some case separated from the dynamic data (e.g. CYME [9] and Eurostag [10] formats).
- *Any number of files.* The Simpow format [5] provides the possibility of including any number of files. Nested file

inclusion is also allowed. This feature is useful in case of large networks, where the amount of data is cumbersome. The user can be interested in modifying only a small part of the network and it is thus easier to work on a small file that is then included in the system.

### 4) The way default values and data manipulation is handled:

- *Default Values.* Most formats, especially fixed position formats, does not support default values and force writing all data of a component, even if those data are not known or could be easily deduced by default.
- *Device Prototyping.* An advanced evolution of default data is the device prototyping. This feature is only provided by the DigSilent format [6]. In this format, any specific device record can refer to a common prototype. This concept is taken from the *class inheritance* of object oriented programming. In case of big distribution networks where most of the transformers and protections are the same, data prototyping can save space. Furthermore a change in the prototype will automatically affect all devices that point to that prototype.
- *Modification Command.* Some formats provide a set of commands for modifying the *base case* data. For example Simpow provides the powerful ALTER command [5]. Matlab-based formats implicitly include modification commands, since any Matlab function and matrix manipulation can be included in the data file. This feature is particularly relevant if one want to study how some data variation affects the system without the need of storing tens of data files.

## B. Data Import/Export Utilities

From the previous section, it is clear that if one wants to use different software tools, one has to use some data import/export utilities. Software companies and open-source groups, such as UWPFLOW [25], InterPSS [23] and PSAT [17], develop and provide adapters to import data. However, the direction is mainly one way, i.e. importing data in different formats into simulation applications, but generally not the other way round, i.e. exporting data to different data formats. This makes very difficult (if not impossible) to exchange power system simulation study case in a robust and reliable way.

Another strong restriction to the diffusion and creation of data format adapters is the fact that the documentation of most commercial data formats is not freely available and, in some cases, is intentionally not complete.

However, even if one could access the full and detailed documentation of any formats in use in the power system community, there is another issue that should be solved, as follows. A simple observation tells us that if there are  $n$  data formats and  $m$  applications, the possible filtering/converting routes that connect each format to each application are  $n \cdot m$ . This fact is pictorially represented in Fig. 1. Actually, the number of routes is even higher, since some formats come in several versions and require a specific filter for each version (e.g. the family of PSS/E v. 2x and 3x formats).

TABLE I  
FEATURES OF MOST COMMON DATA FORMATS FOR POWER SYSTEM ANALYSIS

Format Name	Data Position	Data Order	Dynamic Data	Market Data	Short Circuit Data	Graphic Data	Custom Data	Number of Files	Default Values	Modification Command
CEPEL	Fixed	Fixed	No	No	No	No	No	Unique	No	No
CYME	Fixed	Fixed	Yes	No	Yes	No	No	Multiple, Fixed	Yes	No
DigSilent	Free	Free	Yes	No	Yes	Yes	No	Unique	Prototypes	No
EPRI/BPA	Fixed	Fixed	Yes	No	Yes	No	No	Unique	Yes	No
Eurostag	Fixed	Fixed	Yes	No	Yes	No	No	Multiple, Fixed	Yes	No
FlowDemo.net	Free	Fixed	No	No	No	Yes	No	Unique	No	No
GE-PSLF	Free	Fixed	No	No	No	No	No	Unique	Yes	No
IEEE CDF	Fixed	Fixed	No	No	No	No	No	Unique	No	No
INPTC1	Fixed	Fixed	No	No	No	No	No	Multiple, Fixed	No	Yes
MatPower	Free	Free	No	Yes	No	No	No	Any	No	No
Neplan	Free	Free	Yes	No	Yes	Yes	Yes	Multiple, Fixed	Yes	No
PowerWorld	Free	Free	No	Yes	Yes	Yes	No	Unique	Yes	No
PSAT	Free	Free	Yes	Yes	No	Yes	Yes	Any	No	Yes
PSS/E	Free	Fixed	Yes	No	Yes	No	Yes	Unique	Yes	No
PST	Free	Free	Yes	No	Yes	No	Yes	Any	No	Yes
Simpow	Free	Free	Yes	No	Yes	Yes	Yes	Any	Yes	Yes
UCTE	Fixed	Fixed	No	No	No	No	No	Unique	Yes	No

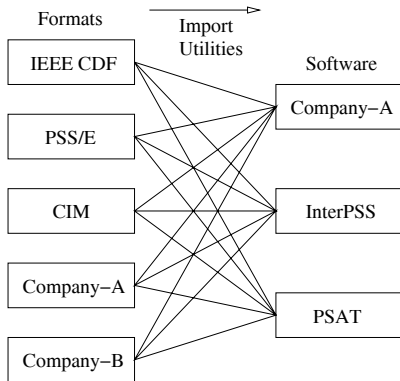


Fig. 1. Current state of data exchange structure.

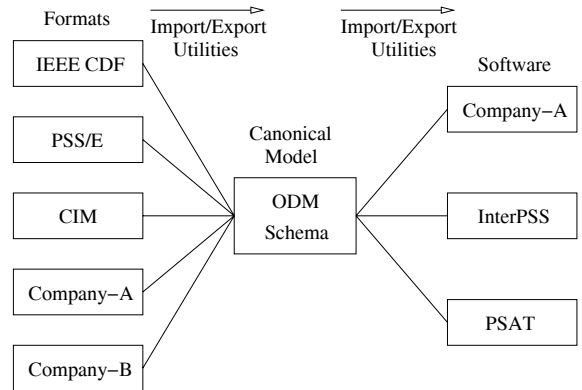


Fig. 2. Proposed data exchange structure.

The proposed ODM is able to solve this issue, given that this schema becomes widely accepted by the power system community and by software companies. In the information model theory, a model like the ODM is sometimes called a *canonical* model. In other words, the ODM is a “super-set” of all other data formats. Using the canonical model, the possible filtering/converting routes for  $n$  formats and  $m$  applications are only  $n + m$  (see Fig. 2).

If the model becomes widely accepted, third-party tools, such as graphic one-line diagram editors and simulation modules, developed based on the ODM, are expected to be provided, mostly by open-source software and research groups (see Fig. 3).

### III. THE OPEN DATA MODEL

The proposed ODM is specified by an XML schema intended for power system simulation and, thus, only represents power systems in a logical view, with nodal bus and branch as the most fundamental concepts. Modern power system simulation algorithms are mostly built on the power network nodal

admittance  $Y$ -matrix. The model is designed to minimize any possible ambiguity to implement the  $Y$ -matrix based solutions. However, the ODM with most elements optional is intended for power system simulation information exchange only, not a power system model for simulation purposes. Thus, the ODM does not represent the  $Y$ -matrix itself.

In the following subsections we present a summary of the basic concepts of the proposed ODM. Complete coverage of the ODM could be found on-line at [26], [27].

#### A. Basic Concepts

1) *Records*: The smallest unit of information is a record. Each record has certain common attributes, such as name, description, status, etc. A minimal power system data file is a set of bus records and branch records, with the branch records “connecting” to two or more bus records to form a network.

2) *Naming Convention*: In addition to the data types defined in XML, there are two main XML data structures used in the ODM: (i) Complex Type and (ii) Simple Type of String. A complex XML type defines a com-

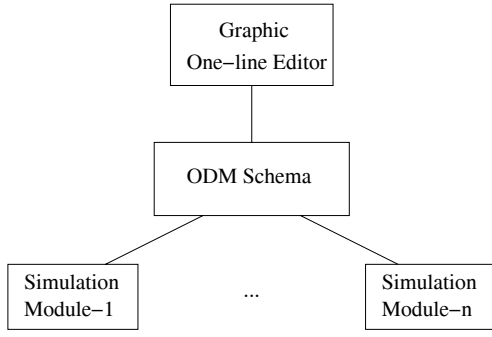


Fig. 3. Tools built around the common data model.

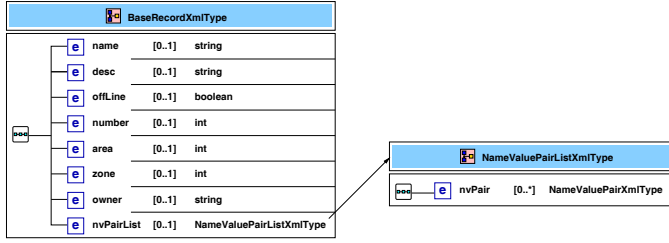


Fig. 4. Prototype of base record data in XML format.

plex data structure with different types of attributes and elements, for example, `PSSNetworkXmlType`. Complex types always end with `XmlType`. A simple XML type of string defines an enumeration of simple tokens, for example, the voltage unit. Simple types of string always end with `DataType`. All types are “qualified” with a prefix `pss`, so that in the future `pss:basePower`, `my:basePower` or `companyA:basePower` could coexist and mean different things.

3) *Units*: International units, such as KV, MVA, and the per unit system can be used in the ODM Model. All per unit values are computed using the system base power rate and the bus base voltage.

### B. Record Attributes

Each record has certain common attributes, such as name, description, status, etc. We use the `BaseRecordXmlType`, which will be the parent of all record types to describe attributes common to all records (see Fig. 4).

All attributes defined are optional, but of course, the more attributes, the more clearly the network is represented. The ODM allows including a name value pair list, where any extra attribute could be defined. For example, one might add the following name-value pair:

```
<pss:nvPairList>
  <pss:nvPair>
    <pss:name>Most Important ODM Feature</pss:name>
    <pss:value>Open, Extensible, created and
      maintained by Open-Source Groups</pss:value>
  </pss:nvPair>
</pss:nvPairList>
```

Most records have a unique identification code or “id”, such as bus records and branch records. At this aim, The

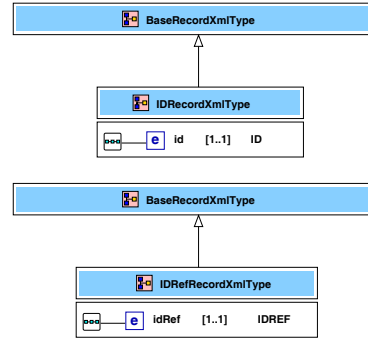


Fig. 5. ID and IDRef record data in XML format.

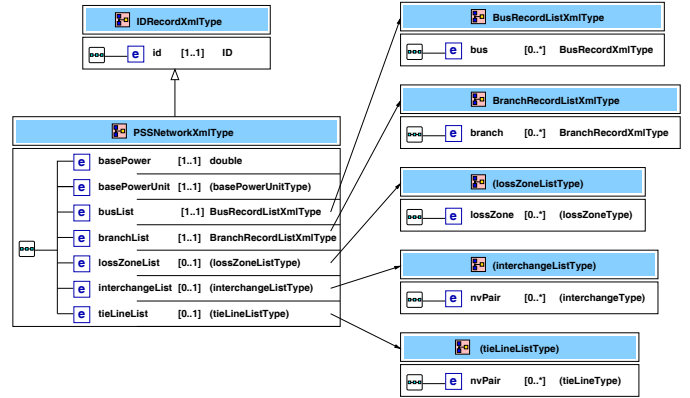


Fig. 6. Network record.

`IDRecordXmlType` defines records with a unique id. The XML schema data type `ID` will make sure that the field is unique within an XML document. Some records reference a certain record in the model. For example, a generator data record references to a bus record, where the generator is connected. The `IDRefRecordXmlType` defines id reference records. The XML schema data type `IDREF` makes sure that the field is referencing an id field defined in the XML document Figure 5 illustrates the `ID` and `IDRef` record data in XML format.

### C. Power System Data

A power system network is represented by the `PSSNetworkXmlType`, which contain a list of bus records and a list of branch records plus other optional records, to represent the basic connection of a power system (see Fig. 6). The bus records are connected (referenced) by the branch records to form a power system. This way of representing a power system is based on an object-oriented power system model that was presented in [28].

1) *Bus Record*: A bus record is a `ID` record with a base voltage (see Fig. 7). Clearly, the bus record can be easily extended to describe more complex data structure for power flow studies, short circuit analysis, etc. The bus record can contain other records defining data of loads, generators, SVCs, etc.

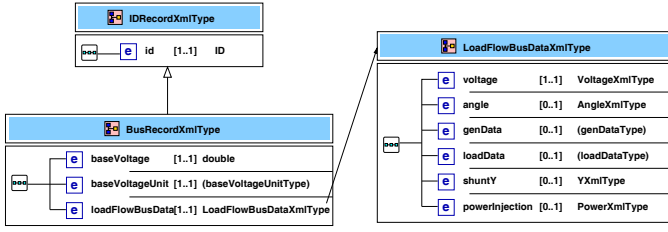


Fig. 7. Bus record.

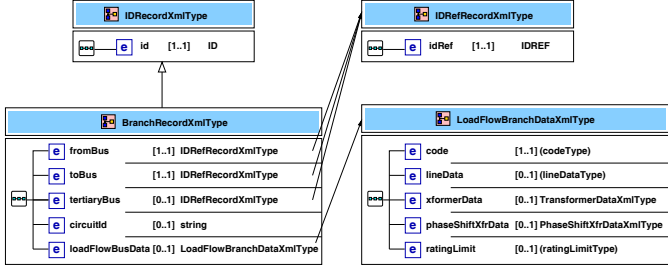


Fig. 8. Branch record.

2) *Branch Record*: Figure 8 shows a two terminal branch record that is a ID record with three additional attributes, namely a *fromBus* reference record, referencing to the *from bus* record; a *toBus* reference record, referencing to the *to bus* record; and a circuit number record that allows distinguishing among multiple parallel branches between two buses. Any other attribute can be included, such as power and voltage rates, series resistances and reactances, tap ratios in case of transformers, etc.

#### D. Case Study

In the analysis of the power system, three key concepts are relevant: (i) base case; (ii) modification to the base case; and (iii) a set of study scenarios. In the ODM, The *Case Study* concept represents all modifications and scenarios of a power system.

1) *Modification*: A list of modification records could be defined and applied to the base case data. Various modifications, including transmission line outage, generator outage etc, could be made to the network either at base case level or at a case study scenario level.

2) *Scenario*: A list of scenario records could be defined for multiple study case scenarios. For example, in N-1 contingency analysis, a particular scenario may be defined by making modification, such as opening one transmission line, to the base case data.

## IV. EXAMPLE

This section illustrates the ODM schema through a simple 5-bus system. The one-line diagram of the 5-bus test system is depicted in Fig. 9. For the sake of completeness, bus and branch data are shown in Tables II and III.

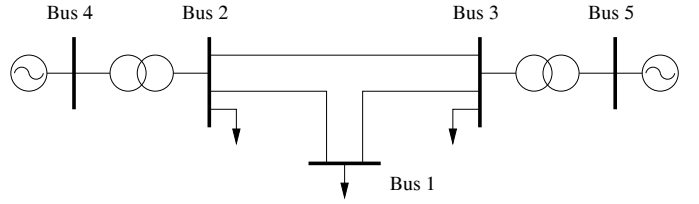


Fig. 9. One-line diagram of the 5-bus test sample.

TABLE II  
5-BUS SYSTEM: BUS DATA.

Bus Name	Bus Type	$V_N$ (kV)	$V$ (p.u.)	$\theta$ (rad)	$P_G$ (MW)	$P_L$ (MW)	$Q_L$ (Mvar)
Bus1	PQ	138	1.0	-	-	160	80
Bus2	PQ	138	1.0	-	-	200	100
Bus3	PQ	138	1.0	-	-	370	130
Bus4	PV	1	1.05	-	500	-	-
Bus5	Swing	4	1.05	0.0	-	-	-

3) *Header Data*: The header of the ODM file defines basic data about the schema and the case study. A basic header is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<pss:PSSStudyCase>
  <pss:id>InterPSS-5Bus-Sample-System</pss:id>
  <pss:schemaVersion>V1.00</pss:schemaVersion>
  <pss:originalFormat>IEEE-ODM-PSS</pss:originalFormat>
  <pss:analysisCategory>Loadflow</pss:analysisCategory>
  <pss:networkCategory>Transmission</pss:networkCategory>
  <pss:baseCase>
    <pss:id>Base-Case</pss:id>
    <pss:basePower>100.0</pss:basePower>
    <pss:basePowerUnit>MVA</pss:basePowerUnit>
    <pss:busList>
      ...
    </pss:busList>
    <pss:branchList>
      ...
    </pss:branchList>
  </pss:baseCase>
</pss:PSSStudyCase>
```

The very first line says that the file is an XML file. The PSSStudyCase type starts the definition of the current case study. A set of tags define the name of the case study (*id*), the version of the schema (*schemaVersion*), the format (*originalFormat*), the kind of analysis are the data for (*analysisCategory*), and the voltage level of network (*networkCategory*).

The *baseCase* type contains the main data of the network. These data can be divided into general data such as the identification code (*id*), and the system power base (*basePower*) and its units (*basePowerUnit*). Bus list and branch lists are the minimum information that defines the network topology. If necessary, also loss zones, interchange areas and tie lines can be included in the base case.

The list of bus (*busList*) and branch (*branchList*) specifications follows. These are described in the following subsections.

4) *Bus Data*: The *busList* type contains the information of each bus of the network. The code for the swing bus is as

TABLE III  
5-BUS SYSTEM: BRANCH DATA.

From Bus Name	To Bus Name	$R$ (p.u.)	$X$ (p.u.)	$1/2B$ (p.u.)	Tap Ratio (p.u./p.u.)
Bus1	Bus2	0.04	0.25	-0.25	-
Bus1	Bus3	0.1	0.35	0.0	-
Bus2	Bus3	0.08	0.3	-0.25	-
Bus4	Bus2	0.0	0.015	0.0	1.05
Bus5	Bus3	0.0	0.03	0.0	1.05

follows:

```
<pss:bus>
  <pss:name>Gen Bus-5 5000 Volt</pss:name>
  <pss:id>Bus5</pss:id>
  <pss:baseVoltage>4000</pss:baseVoltage>
  <pss:baseVoltageUnit>VOLT</pss:baseVoltageUnit>
  <pss:loadflowBusData>
    <pss:voltage>
      <pss:voltage>1.05</pss:voltage>
      <pss:unit>PU</pss:unit>
    </pss:voltage>
    <pss:angle>
      <pss:angle>0.0</pss:angle>
      <pss:unit>DEG</pss:unit>
    </pss:angle>
    <pss:genData>
      <pss:code>SWING</pss:code>
    </pss:genData>
  </pss:loadflowBusData>
</pss:bus>
```

The code for the PV bus is as follows:

```
<pss:bus>
  <pss:name>Gen Bus-4 1000 Volt</pss:name>
  <pss:id>Bus4</pss:id>
  <pss:baseVoltage>1000</pss:baseVoltage>
  <pss:baseVoltageUnit>VOLT</pss:baseVoltageUnit>
  <pss:loadflowBusData>
    <pss:voltage>
      <pss:voltage>1.05</pss:voltage>
      <pss:unit>PU</pss:unit>
    </pss:voltage>
    <pss:genData>
      <pss:code>PV</pss:code>
      <pss:gen>
        <pss:p>500.0</pss:p>
        <pss:q>0.0</pss:q>
        <pss:unit>MVA</pss:unit>
      </pss:gen>
    </pss:genData>
  </pss:loadflowBusData>
</pss:bus>
```

The code for the PQ bus is as follows:

```
<pss:bus>
  <pss:name>Bus-1 138.0 KV</pss:name>
  <pss:id>Bus1</pss:id>
  <pss:baseVoltage>138</pss:baseVoltage>
  <pss:baseVoltageUnit>KV</pss:baseVoltageUnit>
  <pss:loadflowBusData>
    <pss:voltage>
      <pss:voltage>1.0</pss:voltage>
      <pss:unit>PU</pss:unit>
    </pss:voltage>
    <pss:loadData>
      <pss:code>CONST_P</pss:code>
      <pss:load>
        <pss:p>160</pss:p>
        <pss:q>80</pss:q>
        <pss:unit>MVA</pss:unit>
      </pss:load>
    </pss:loadData>
  </pss:loadflowBusData>
</pss:bus>
```

```
</pss:loadData>
</pss:loadflowBusData>
</pss:bus>
```

5) *Branch Data*: The branchList type contains the information of each bus of the network. The code for a transmission line branch is as follows:

```
<pss:branch>
  <pss:id>Bus1_to_Bus2_cirId_1</pss:id>
  <pss:fromBus>
    <pss:idRef>Bus1</pss:idRef>
  </pss:fromBus>
  <pss:toBus>
    <pss:idRef>Bus2</pss:idRef>
  </pss:toBus>
  <pss:circuitId>1</pss:circuitId>
  <pss:loadflowBranchData>
    <pss:code>Line</pss:code>
    <pss:lineData>
      <pss:z>
        <pss:r>0.04</pss:r>
        <pss:x>0.25</pss:x>
        <pss:unit>PU</pss:unit>
      </pss:z>
      <pss:totalShuntY>
        <pss:b>0.5</pss:b>
        <pss:unit>PU</pss:unit>
      </pss:totalShuntY>
    </pss:lineData>
  </pss:loadflowBranchData>
</pss:branch>
```

The code for a basic two-winding transformer branch is as follows:

```
<pss:branch>
  <pss:id>Bus3_to_Bus5_cirId_1</pss:id>
  <pss:fromBus>
    <pss:idRef>Bus3</pss:idRef>
  </pss:fromBus>
  <pss:toBus>
    <pss:idRef>Bus5</pss:idRef>
  </pss:toBus>
  <pss:circuitId>1</pss:circuitId>
  <pss:loadflowBranchData>
    <pss:code>Transformer</pss:code>
    <pss:xformerData>
      <pss:z>
        <pss:r>0.0</pss:r>
        <pss:x>0.03</pss:x>
        <pss:unit>PU</pss:unit>
      </pss:z>
      <pss:fromTurnRatio>1.05</pss:fromTurnRatio>
    </pss:xformerData>
  </pss:loadflowBranchData>
</pss:branch>
```

One may claim that the XML schema is not easy to understand for a human reader and that it produces large files even for small networks. This is certainly true but not a real issue because XML files are typically handled by dedicated software applications that make easy organize and visualize the data contained in the file. As matter of fact, XML is widely used in most engineering and scientific fields, including medicine. Actually, one of the few fields where XML is not currently used is power engineering.

## V. ADAPTER IMPLEMENTATION

This section describes a variety of implementations of adapters that converts data files to or from the ODM format.

6) *ODM-PSAT and PSAT-ODM Adapters*: PSAT allows to import and export data in the ODM format [29], [30]. The functions that do the import/export conversion are `odm2psat` and `psat2odm.m`, respectively. Both filters are modular and can be extended as the ODM schema grows.

7) *ODM-InterPSS Adapter*: A plugin has developed to import data in the ODM format into InterPSS for power flow analysis [23].

8) *Other Adapters*: Adapters to convert data from the IEEE CDF and the UCTE format are currently available in the reference implementation. Adapters for PSS/E, GE-PSFL and BPA data formats are currently under development.

## VI. CONCLUSIONS

The paper has presented a novel approach based on the XML schema for storing and exchanging power system data. We call this approach the Open Data Model. It is “open” for two reasons: because it is freely documented and because it can be easily adapted to include any information related to power system analysis.

The implementation of the ODM schema is currently at an early development stage. Next phases include but are not limited to the definition of dynamic data, market data and the implementation of a graphic one-line editor to quickly create and edit ODM documents.

## REFERENCES

- [1] Working Group on a Common Format for Exchange of Solved Load Flow Data, “Common Format for the Exchange of Solved Load Flow Data,” *IEEE Transactions on Power Apparatus and Systems*, vol. 92, no. 6, pp. 1916–1925, Nov./Dec. 1973.
- [2] “Extensible Markup Language (XML),” available at <http://www.w3.org/XML>.
- [3] E. Z. Zhou, “XML and Data Exchange,” *IEEE Power Engineering Review*, pp. 66–68, Apr. 2000.
- [4] “Siemens PTI (Power Technologies International),” available at <http://www.pti-us.com>.
- [5] “ABB Simpov,” available at <http://www.stri.se>.
- [6] “DigSilent, Power System Engineering and Software,” available at <http://www.digsilent.de>.
- [7] “NEPLAN,” available at <http://www.neplan.ch>.
- [8] G. Energy, *GE-PSLF Load Flow Data Export/Import File for PSLF Version 15.1*, General Electric International, Inc., 1 River Road, Schenectady, NY 12345, USA, June 2005.
- [9] “CYME International Inc.” available at <http://www.cyme.com>.
- [10] “Eurostag,” available at <http://www.eurostag.be>.
- [11] PowerWorld Corporation, *POWERWORLD Simulator Version 13.0: Interactive Power System Simulator Analysis and Visualization*, Powerworld Corporation, available at <http://www.powerworld.com>, 2008.
- [12] “ANA’s Software,” available at <http://www.cepel.br/servicos/descprog.shtm>.
- [13] J. Chow, *Power System Toolbox Version 2.0: Load Flow Tutorial and Functions*, Cherry Tree Scientific Software, RR-5 Colborne, Ontario K0K 1S0, 1991-1999.
- [14] —, *Power System Toolbox Version 2.0: Dynamic Tutorial and Functions*, Cherry Tree Scientific Software, RR-5 Colborne, Ontario K0K 1S0, 1991-1997.
- [15] “Enel,” available at <http://www.enel.com>.
- [16] “Union for the Co-ordination of Transmission of Electricity (UCTE),” available at <http://www.ucte.org>.
- [17] F. Milano, “PSAT, Documentation the Power System Analysis Toolbox version 2.0.0,” 2007, available at <http://www.uclm.es/area/gsee/Web/Federico/psat.htm>.
- [18] R. D. Zimmerman and D. Gan, *Matpower, Documentation for version 2*, Power System Engineering Research Center, Cornell University, 1997, available at <http://www.pserc.cornell.edu/matpower/matpower.html>.

- [19] “Power Education Toolbox (PET),” available at <http://www.ece.neu.edu/~abur/pet.html>.
- [20] “Extended Transient-Midterm Stability Package: User’s Manual for the Power Flow Program,” Jan. 1987, ePRI computer code manual EL-2002-CCM.
- [21] “FlowDemo.net,” (EEH - Power Systems Laboratory, Zürich. Available at <http://flowdemo.net>.
- [22] PowerWorld Corporation, *Auxiliary File Format for Simulator 11.0*, Powerworld Corporation, available at <http://www.powerworld.com>, Apr. 2005.
- [23] M. Zhou, “InterPSS,” available at <http://www.interpss.org>.
- [24] J. Chow, “Power System Toolbox,” 2002, available at <http://www.eagle.ca/~cherry>.
- [25] C. A. Cañizares and F. L. Alvarado, “UWPFLOW Program,” 2000, university of Waterloo, available at <http://www.power.uwaterloo.ca>.
- [26] M. Zhou, “ODM Data Structure and Relationship,” available at <http://sites.google.com/a/interpss.org/interpss/Home/ieee-pes-oss/data-structure-relationship>.
- [27] —, “ODM Schema Specification,” available at <http://sites.google.com/a/interpss.org/interpss/Home/ieee-pes-oss/xml-schema-specification>.
- [28] E. Z. Zhou, “Object-oriented Programming, C++ and Power System Simulation,” *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 206–216, Feb. 1996.
- [29] F. Milano, “An Open Source Power System Analysis Toolbox,” *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1199–1206, Aug. 2005.
- [30] F. Milano and L. Vanfretti and J. C. Morataya, “An Open Source Power System Virtual Laboratory: The PSAT Case and Experience,” *IEEE Transactions on Education*, vol. 51, no. 1, pp. 17–23, Feb. 2008.



**Federico Milano** (M’03) received from the University of Genoa, Italy, the Electrical Engineering degree and the Ph.D. degree in 1999 and 2003, respectively. From 2001 to 2002 he worked at the University of Waterloo, Canada, as a Visiting Scholar. He is currently an assistant Professor at the University of Castilla-La Mancha, Ciudad Real, Spain. His research interests include voltage stability, electricity markets and computer-based power system analysis and control.



**Mike Zhou** (M’90), Ph.D. Tsinghua University, 1987. Currently a senior computer system architect with TIBCO Software, Inc., Palo Alto, CA, USA. He is also the architect of the InterPSS project.



**GuanJi Hou** received the B.S. degree in electrical engineering from Electric Power College, South China University of Technology, Guangdong, China, in 2006. Since then, he has been studying at the South China University of Technology for the M.S. degree. Currently, he is a member of InterPSS development team. His main field of interest includes computer-based and especially grid-computing-based power system simulation and analysis.