ON CREATING A NEW FORMAT FOR POWER QUALITY AND QUANTITY DATA INTERCHANGE

Erich W. Gunther
Electrotek Concepts, Inc.
Knoxville, Tennessee
erich@electrotek.com

Abstract -- **The recent release of IEEE 1159, A Recommended Practice for Monitoring Power Quality [1] has resulted in a consistent set of terms, definitions and guidelines for monitoring power quality. IEEE 1159 does not address the need however, for a consistent mechanism for transporting the output of these instruments to the end user. Each vendor has their own communications, control, and analysis software that is not compatible with any other vendor.**

In addition to the difficulty with exchangeability of monitored data, there is a similar difficulty in exchanging and comparing the results of computer simulations. Simulation tool vendors also use a variety of file formats and conventions appropriate for their programs.

Given that today's power quality engineers need to transport the output of different vendors monitors and simulation programs to various database and analysis programs (also from multiple vendors), a need exists for a standard data interchange format. This paper describes the principal issues that need to be addressed in the development of such a standard. A brief description of an interchange format developed to test the assertions made in this paper is also provided.

Keywords -- Power Quality, Measure, Monitor, Simulate, Data, File, Interchange, Format

I. INTRODUCTION

A power quality engineer uses several tools in the investigation of a power quality problem. These tools can be roughly classified as information sources, data transport, data storage, and data analysis. Although there are some feedback loops not shown in the diagram, Figure 1 illustrates this basic flow of information.

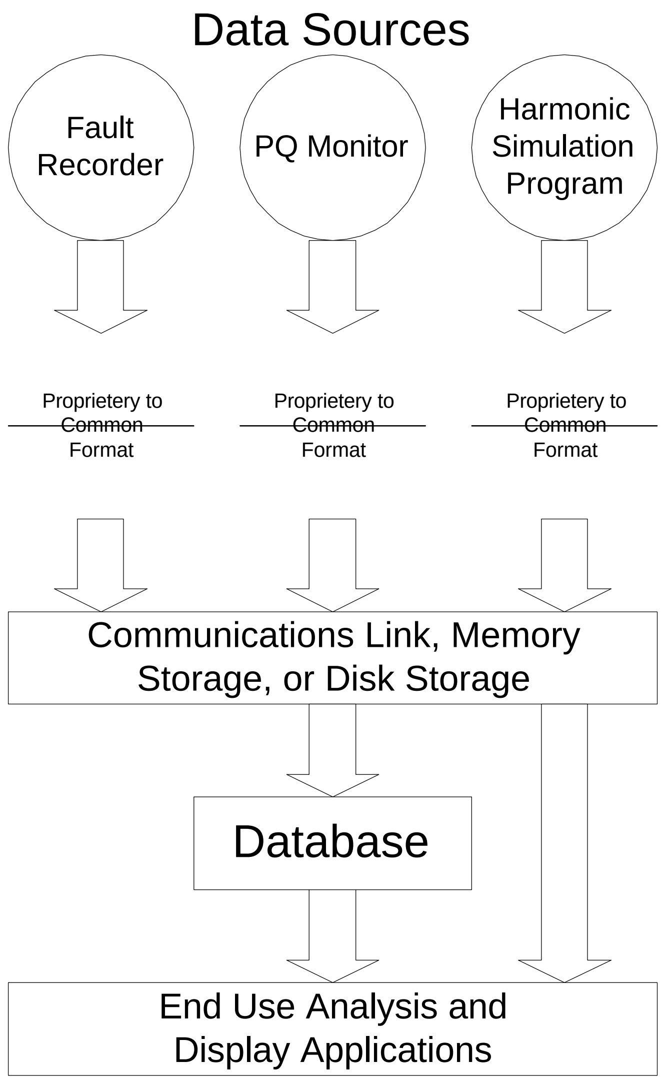

Figure 1 - Information Flow Diagram

As Figure 1 indicates, there are two principal components between the source of the data and the analysis and eventual consumption of the data. These components are the data interchange component and the database component.

The database component is an important part of a complete power quality data management system, especially for large projects, long term surveys, and archival storage of data. Recent publications have discussed the principal attributes and demonstrated implementations of such databases [2]. This paper does not address this component.

The data interchange component is the subject of this paper. Without it, the other components are limited in their usefulness. Traditionally this component has been implemented through the use of proprietary and/or vendor specific file formats and communications protocols. Often, especially in the case of small or simple systems, the database component is not even present (or necessary). In these cases, the interchange file itself acts as the archival medium.

A similar need to exchange multi-vendor data was determined several years ago by the IEEE relaying committee for the exchange of fault recorder data. This resulted in the IEEE COMTRADE format [3]. This format provides for an ASCII or binary file format for the interchange of transient data produced by fault recording devices. This format is quite useful for general time domain data interchange and is in fact in use by the power quality community. A number of vendors of monitoring instruments, simulation programs, and analysis programs have provided the means to read and write this format to facilitate the exchange of time domain data, be it measured or simulated [4].

As useful as IEEE COMTRADE is, it was necessarily limited in scope due to the scope of its originating committee. The principal limitation is its inability to deal with frequency domain and probability domain information. Also, it is somewhat limited in its extensibility and the ability to include additional site, instrument, and vendor specific information associated with a measurement or simulation.

II. INTERCHANGE REQUIREMENTS

Through the course of developing and using various measurement and simulation systems, the author has developed a list of requirements that seem to be important for inclusion in a data interchange format.

A data interchange format must accommodate the following fundamental types of data sources:

- Measuring Devices
- Computer Simulated Data
- Manually Recorded Data

This requirement represents the fundamental need of an interchange format to facilitate the comparison of data from multiple sources. For example, Figure 2 illustrates a comparison of a voltage measurement from a power quality monitor and a simulation of the same system from a harmonic simulation program.

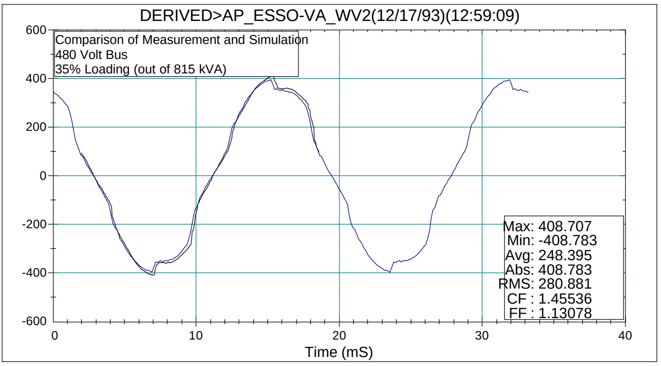

Figure 2 - Measurement / Simulation Comparison

Each of the data sources mentioned above can generate the following types of observations:

- Quantity versus Time
- Quantity versus Frequency
- Quantity versus Probability
- Any of the above versus Time

An observation consists of one or more simultaneously measured, simulated, or recorded quantities (data channels). These quantities are usually recorded as a series of numbers. One or more series are needed to represent the data. At least one of these series represents the dependent variable. Others may represent additional dependent or independent variables needed to quantify the item. For example, measurement instruments would directly associate a magnitude series with a channel. A simulation program would associate a simulated waveform for a bus voltage with a channel. A single channel can therefore result in one or more series depending on the instrument and measured quantity. The term “data channel” is used to represent this aggregate object in this paper.

The magnitude of a quantity that is represented as a series can come in a variety of forms:

- Value (Instantaneous Magnitude)
- RMS Magnitude
- Phasor (RMS Magnitude and Phase)

These forms have proved to be sufficient for representing a wide variety of data obtained from various simulation tools and measurement instruments. The following are possible quantities that can be represented using one or more series:

- Waveform - Time Series, Value Series
- Spectrum - Frequency Series, Phasor Series
- System Stability - Time Series, Phasor Series
- RMS Variation - Time Series, RMS Magnitude Series
- Voltage Fluctuation (Flicker) - Time Series, RMS Magnitude Series
- IEC Flicker - Time Series, Value Series (Pst)
- IEC Flicker (Processed) - Probability Series, Value Series
- Frequency Response - Frequency Series, Phasor Series
- Energy - Time Series, Value Series

For any quantity, there are often a number of variants of that quantity recorded (or simulated) at each sampling interval. Each variant requires a series in addition to the series for the independent variable (e.g. time series). Examples of variants available for a single data channel include:

- Value
- Min/Max
- Min/Max/Avg
- Min/Max/Inst
- Min/P5/P10/Avg/P90/P95/Max

The P5, P10, P90 and P95 notations represent the values at 5%, 10%, 90%, and 95% probabilities from a cumulative probability function (CPF). This method of quantifying a result is becoming more common as probabilistic methods are used to describe the results of simulations and measurements. This is the central method used in the electromagnetic compatibility (EMC) approach adopted by the IEC [5].

As mentioned previously, a quantity is represented by a series of numbers. These series may be represented in a variety of ways:

- Simple array of values
- Array of values with series scale factors
- Start value, increment, number of values
- Start value, array of incremental values

The choice of series storage representation depends on a variety of attributes including data source type, implementation technology, and storage requirements. The values in a series may be real or complex.

In addition to the basic requirements and attributes outlined above, there are numerous pieces of supporting information that must accompany a data source. This information includes naming, location, setup, labeling, units, and other data. Also, a means should be provided for an implementor to add private data to the file that can be used by programs that are aware of it, and ignored by those that are not. In other file formats this is sometimes referred to as hint information (e.g. the Microsoft True Type font file format).

The following tables illustrate some of the properties that can be associated with each component object in a data interchange schema. The descriptions that follow are based in a top down decomposition of the objects that make up the interchange data (shown graphically in Figure 3).

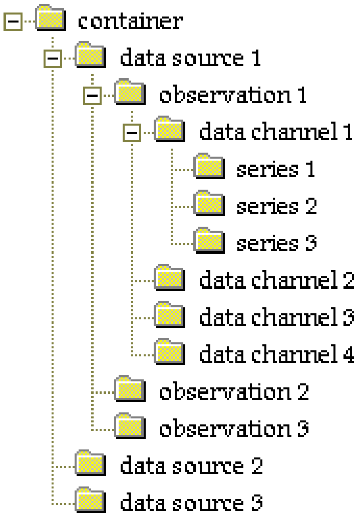

Figure 3 - Data Interchange Objects

A data interchange schema may be implemented in a variety of containers:

- Disk based file - ASCII or BINARY, Conventional or Structured
- Memory object
- Serial communication stream

A container is a convenient place for the data to be transferred to reside between source and destination. A container can have a number of attributes which should include:

- Creation Time
- Author
- Owner
- Access Control List
- Size
- Type
- One or more data sources
- Private Data

A container could hold one or more data sources. These data sources represent the output of a monitoring device, simulation program, or some manually entered data. Two different data sources might exist that differ only in the setup of the instrument, or the configuration of a simulation program. The following attributes can be associated with such a data source:

- Name
- Type - Measure, Manual, Simulate
- Vendor
- Location Description
- Location Lat./Long.
- Install time
- Removal or setup change time
- One or more physical data channel descriptions
- One or more observations
- Vendor specific setup information
- System configuration information
- Private Data

Each data source can have one or more data channels. These data channels may be physical as in the case of an instrument or logical as in the case of a simulation program. In any case a data channel could have the following attributes:

- Channel Number
- Name
- Location Identifier
- Phase or Other Name - a, b, c, ab, bc, neutral, residual, pos. seq.
- Channel Type - Time Domain, Freq. Domain, XY Data, Probability
- Channel Sub Type - Scan, Spectrum, Correlation, General, CPF, Histogram
- Quantity Type (Voltage, Current, Power, Energy, Temperature, Frequency)
- Quantity Units (Volts, Amps, Watts, HP, Joules, Ergs, C, F, Cycles, Hertz)
- Preferred Greek Prefix (automatic, milli, micro, etc.)
- Base Quantity (for per-unitization)
- Preferred units mode - Per Cent, PU, Eng. Units
- Variant Style (described below) - value, min./max., min./max./avg., etc.
- Trigger Type - not triggered, high level, low level, float
- High Trigger Level
- Low Trigger Level
- Vendor specific setup information
- Private Data

The location identifier listed above is intended to represent the name of the specific point to which the data from that channel is obtained. For example, a measurement devices that records 16 channels might be used to monitor two different voltage busses and some feeder currents. The location identifier for channel 1 might then be “Transformer 5402 High Side Bus”, and the phase name “A”.

In addition to the description of the attributes of each physical or logical data channel, a data source object contains a list of one or more observations made using those channels. These observations might have the following attributes:

- Name
- Create Time
- Start Time
- Trigger Time
- Serial Number
- One or more simultaneously measured, simulated, or recorded quantities (data channels)
- Private Data

As discussed earlier, any given observation contains information measured, simulated or recorded on one or more data channels. Each data channel recording therefore has a number of attributes that can be associated with it:

- Channel ID
- One or more series
- Default Display Value - Magnitude, Phase, Real, Imaginary
- Private Data

Depending on the channel type, sub-type, and variant, one or more series of data points are saved. Each series object can have the following properties:

- Quantity Type (Voltage, Current, Power, Energy, Temperature, Frequency)
- Quantity Units (Volts, Amps, Watts, HP, Joules, Ergs, C, F, Cycles, Hertz)
- Base Quantity (for per-unitization)
- Private Data

The quantity type and units associated with a series may override the same information that is provided as part of the data channel information. The data channel information is intended to indicate the type and units of the principal series. For some applications that generate XY type data (primarily simulation programs), then the series types and units information are more significant to the end use application.

III. CONTAINERS

The information provided here lays the groundwork for specifying a complete data interchange format.

As indicated earlier, the basic schema can be implemented in a variety of containers. There is a class of container that is of particular note given today's popular computing environments. This container is the structured storage system that is part of Microsoft's Object Linking and Embedding (OLE) Version 2.0 specification.

OLE structured storage provides a common method of storing data in memory, a communications stream, or in memory for applications running in any of Microsoft's Windows environments. This storage model provides a transaction based (with rollback) mechanism for storing data and a file system within the file system. The equivalent of sub-directories and files can be created within a single disk file or memory object. This mechanism simplifies the management of data broken down into sets of aggregate objects like the data schema described in this paper.

Use of structured storage also simplifies the data file implementation of an interchange format by having all components of the interchange data in a single file. This allows implementers of software that manipulate interchange files to work with program generators like those from Microsoft and Borland that assume a single data file is involved when using common user interface elements like open and save dialog boxes. This has been a nuisance with formats like IEEE COMTRADE, for example, which has three files that make up the interchange format.

The interchange format concepts described in this paper of course do not require the use of structured storage. Traditional ASCII and binary representations of the schema can be used. These methods rely on the use of tags (in the case of ASCII files) or sizes and offsets (for binary files) to segregate the various objects in the file. Particular attention must be paid to speed of access when devising these schemes however.

CONCLUSIONS

A set of requirements and desirable attributes for a power quality data interchange format have been described. Key among these is the ability to represent data from a variety of sources (measured, simulated, or manually created), in the time, frequency, and probability domains. An example of a file format with many of these attributes has been developed and is being used in a research environment. This format has been put into the public domain [6] by the author to facilitate debate and development of an eventual standard.

REFERENCES

[1] IEEE P1159, A Recommended Practice for Monitoring Power Quality, Draft 6a, June 1995.

[2] Probing Power Quality Data, IEEE Computer Applications in Power, Volume 7, Number 2, April 1994

[3] IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems, IEEE C37.111-1991, October 21, 1991

[4] Visualization of Power System Data Using a PC-Based GUI Interface, presented at the First European Conference on Power System Transients, Lisbon, Portugal, June 17-18, 1993.

[5] IEC 50(161): 1990, International Electrotechnical Vocabulary (IEV), Chapter 161: Electromagnetic Compatibility.

[6] The PQ Network, Electrotek Concepts, Inc., http://www.pqnet.electrotek.com/.

BIOGRAPHY

Erich W. Gunther graduated from Gannon University in 1980 and from Rensselear Polytechnic Institute in 1984 with a Masters of Electric Power Engineering degree. From 1980 to 1984 he worked in the broadcast television industry. In 1984 he joined McGraw-Edison's (now Cooper Power) Systems Engineering group where he developed a variety of computer simulation programs including the initial version of the V-HARM harmonic simulation program. In 1988 Mr. Gunther and two other associates left Cooper Power Systems and formed the power system engineering group at Electrotek Concepts, Inc. Since joining Electrotek, Erich has been involved in a variety of projects including developing the data acquisition system and methodology for the EPRI distribution power quality project, developing the SuperHarm harmonic simulation program, developing TOP, The Output Processor, and participating in the development of the Square D power logic system software. Mr. Gunther is currently responsible for technology development at Electrotek and is working on power monitoring systems that take advantage of the national information infrastructure (the Internet). Erich is a member of IEEE and CIGRE and serves on several working groups including acting as the chairman of the CIGRE/CIRED 36.05/CC02 joint working group on voltage quality.