Open Source Fault Location in the Real World

Fred Elmendorf Grid Protection Alliance Inc.
and
Theo Laughner Tennessee Valley Authority

2013 Georgia Tech Fault and Disturbance Analysis Conference
May 7, 2013

Table of Contents

ABSTRACT

Accurate and timely fault location on the power system is critical to the reliability and availability of the bulk electric supply, and many techniques have been developed to address this need. Most substation IEDs have built-in features that provide distance to fault reports, power system engineers the world over have combined science and art to create their own unique solutions, and software vendors have offered commercial solutions. In spite of all of this effort, very few electric utility companies have effective enterprise wide automated fault location systems. Using the open source software approach to solving this problem creates opportunities that were not previously available. The open source fault location engine (openFLE) is a project designed to be flexible, extensible, and adaptable, facilitating deployments ranging from a desktop tool for individual engineers, all the way to enterprise systems automatically processing event records as soon as they are available from field devices. Furthermore, it is built as a platform where there is no restriction on the number or complexity of the algorithms used for fault detection, identification, and distance calculations, and can be extended to support other types of event related data analysis. Grid Protection Alliance (GPA) is a nonprofit free open source software (FOSS) developer specializing in applications for the electric utility industry. GPA builds collaborative relationships among government, regulators, vendors, grid owners, grid operators, research organizations, and academia, with a mission to improve the reliability and resiliency of the electric grid. The openFLE project leverages EPRI's research and GPA's development to provide TVA with an opportunity to improve their operations through the application of new technology. This paper looks at the open source landscape, with a focus on the electric utility needs, and provides a case study for TVA's real-world use of the openFLE. It will also present a history of open source software, including applications for electric utilities, complete with website references, and will provide examples of current open source projects in research, industry, and government.

INTRODUCTION

While we are not attempting to present an exhaustive and authoritative reference manual for open source software, it is important to have a basic understanding of the history, and landscape, so that you can appropriately evaluate the concepts presented. A major goal of open source software is to enable a living and thriving community facilitating rapid development and deployment of high quality software solutions, and continuous improvement and evolution. To convey the power of free open source software as a solution for real world problems, two examples are provided in this paper. The first one, openFLE, is a research and development project to show a proof of concept. The second one, openEDA, is taking that proof of concept and applying it in a fully functional pilot system in a large utility. We also discuss the role of standards in free open source software applications, and conclude with a simple recipe for an open source fault location system.

BRIEF HISTORY AND WEB SITES

Open source software has been a part of the computer evolution since the beginning, but by the mid to late 1970s computer manufacturers had discovered that they could protect their interests, limit competition, and even differentiate themselves through the proprietary software, while generating a new source of revenue. Though this was good for the computer manufacturers, it severely limited the ability to innovate, and some literature refers to it as having a decimating effect on research. Software applications for the electric power system were not exempt from this trend, and by the late 1980s, the proprietary landscape was being accepted as the "norm". Even though proprietary software has overtaken many aspects of the computing universe, open source software has never ceased to exist, and has taken on a variety of forms. In the early days it was referred to as free software, but that term was ambiguous, since it could mean either "Free! As in Free Beer" or "Free! As in NO COST". In 1998, the Open Source Initiative coined the term "open source" as a marketing strategy to remove negative connotations and ambiguity. However, there continues to be a distinction between "open source software" (OSS)and "free open source software" (FOSS). OSS indicates that the source code is open to obtain, review, and modify under certain conditions, but does not necessarily mean that it is free. FOSS indicates that in addition to the benefits of OSS, there is no initial purchase price. Unfortunately, these simple explanations fall short, since there are hundreds of licensing options for OSS and FOSS. The important point is that having the source code available to stimulate innovation and extension provides great advantages. In contrast, the overview section of the website **Open Source Software for Electricity Market Research, Teaching, and Training** has this to say: "lack of open-source access prevents users from gaining a complete and accurate understanding of what has been implemented, restricts the ability of users to experiment with new software features, and hinders users from tailoring software to specific needs. In addition, these packages can be cumbersome to use for research, teaching, and training purposes requiring intensive experimentation and sensitivity analyses".

Below, is a list of interesting OSS/FOSS web sites, ranging from general information, to government sites, and electric utility related sites.

- "Producing Open Source Software – How to Run a successful Free Software Project" http://www2.econ.iastate.edu/tesfatsi/ProducingOSS.KarlFogel2005.pdf
- Free Software Foundation - http://www.fsf.org/
- The Open Source Initiative (OSI) - http://opensource.org/
- OSS/FS - http://www.dwheeler.com/oss_fs_why.html#appendix
- Open Source Software for Electricity Market Research, Teaching, and Training – http://www2.econ.iastate.edu/tesfatsi/electricoss.htm

- DOE Policy - http://science.energy.gov/~/media/ascr/pdf/research/docs/Doe_lab_developed_software_policy.pdf
- The White House Developers - http://www.whitehouse.gov/developers
- IEEE Task Force on OSS for Power Systems - http://ewh.ieee.org/cmte/psace/CAMS_taskforce/software.htm
- UWPFLOW - https://ece.uwaterloo.ca/~ccanizar/software/pflow.htm
- TEFTS - http://thunderbox.uwaterloo.ca/~claudio/software/tefts.htm
- MATPOWER - http://www.pserc.cornell.edu/matpower/
- PSAT - http://www.uclm.es/area/gsee/Web/Federico/psat.htm
- InterPSS - http://www.interpss.org/
- AMES - http://www.econ.iastate.edu/tesfatsi/AMESMarketHome.htm
- DCOPFJ - http://www.econ.iastate.edu/tesfatsi/DCOPFJHome.htm
- OpenDSS - http://sourceforge.net/projects/electricdss/develop

- MatDyn - http://www.esat.kuleuven.be/electa/teaching/matdyn/
- GPA Projects – http://www.gridprotectionalliance.org

VALUE OF FREE OPEN SOURCE

There are many types of value that can be realized through the use of open source software. Some are related to innovation and change, some are financial, and others are related to quality and security. The following list is presented as an overview, in no particular order. As you read through the list, you can expect to think of others that should be added, and you can see that they apply in the electric utility industry, just as any other.

- Reduces barriers for adoption
- Improves flexibility with no vendor lock-in
- Reduces development and maintenance costs
- Reduces total cost of ownership
- Transparency assures high quality, no way to hide bugs
- Perfect platform for collaboration
- Ability to keep improving through new use cases
- Facilitates rapid deployment of state-of-the-art techniques
- Facilitates innovation and encourages extension and refinement
- Community evaluation lowers risk (especially security risk) and improves quality
- Adapts quickly to new threats, standards, or business needs
- Reduces time to market, and can provide “first-mover” advantage
- Greater return on investment for both private and public funds
- Provides an avenue for broader public benefit from government investments and projects

A NEW APPROACH TO FAULT LOCATION

Since every digital fault recorder (DFR), and most other intelligent electronic devices (IEDs) in transmission substations provide the ability to perform distance to fault calculations, it may seem curious that yet another fault location strategy would be necessary. However, in many utilities the existing processes involve multiple desktop tools, one from each different hardware vendor, and tedious manual steps in the data analysis and integration required to obtain a fault distance. Additionally, the accuracy of the fault distance calculation is dependent on the accuracy of system parameters stored in the remote device, and the skill of the engineer performing the analysis, among many other variables.

In 2012, the open source fault location engine (openFLE) was created for TVA through an EPRI project. The openFLE was designed as a platform to accept system configuration parameters through an XML file, parse COMTRADE or PQDIF files as they were placed in the appropriate file folder, and write the fault distance and 30 other calculated values to specified files in an output folder. See Figure 1 for the flow of files through the openFLE demonstration system.

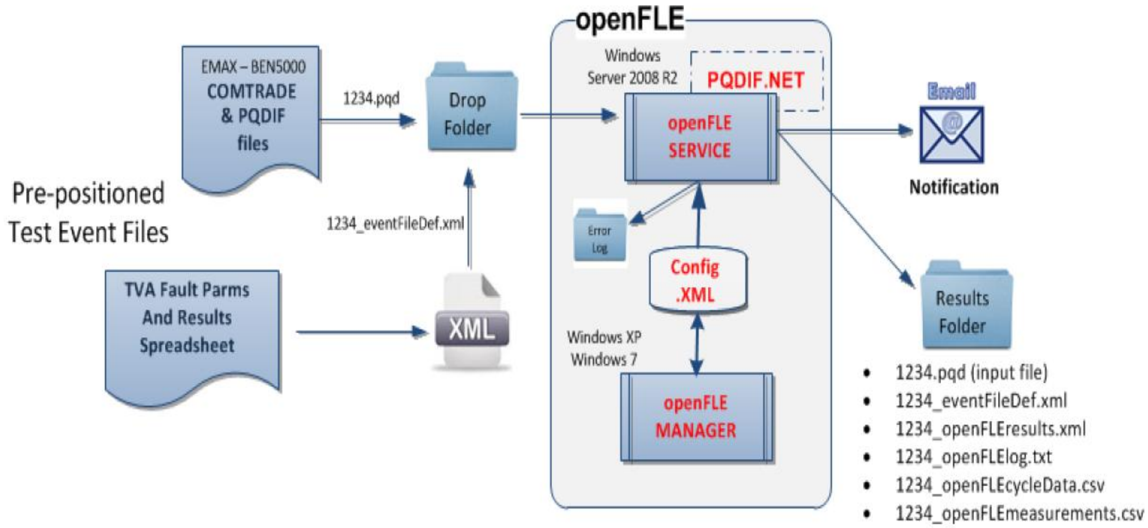

Figure 1 Flow of Files in openFLE

The openFLE was a research and demonstration project, to show proof of concept in automatically analyzing fault records, and there were a number of conditions that had to be met, and assumptions that had to be true, for the openFLE to run properly and produce accurate and consistent results. For example, each event record, whether from a DFR or a PQ monitor, was expected to contain event data from one line. An associated and appropriately named system parameter file was required to be present in the input folder along with the event record, the event record was required to have a full cycle of pre-fault data, and an arbitrary current threshold was specified to identify the inception of a fault. Even though the openFLE is designed as a platform with the ability to accommodate any number of algorithms to identify fault presence, type, and distance, in this demonstration project it was implemented with one simple text book algorithm in each category. See Figure 2 for the steps involved in processing event files using openFLE.

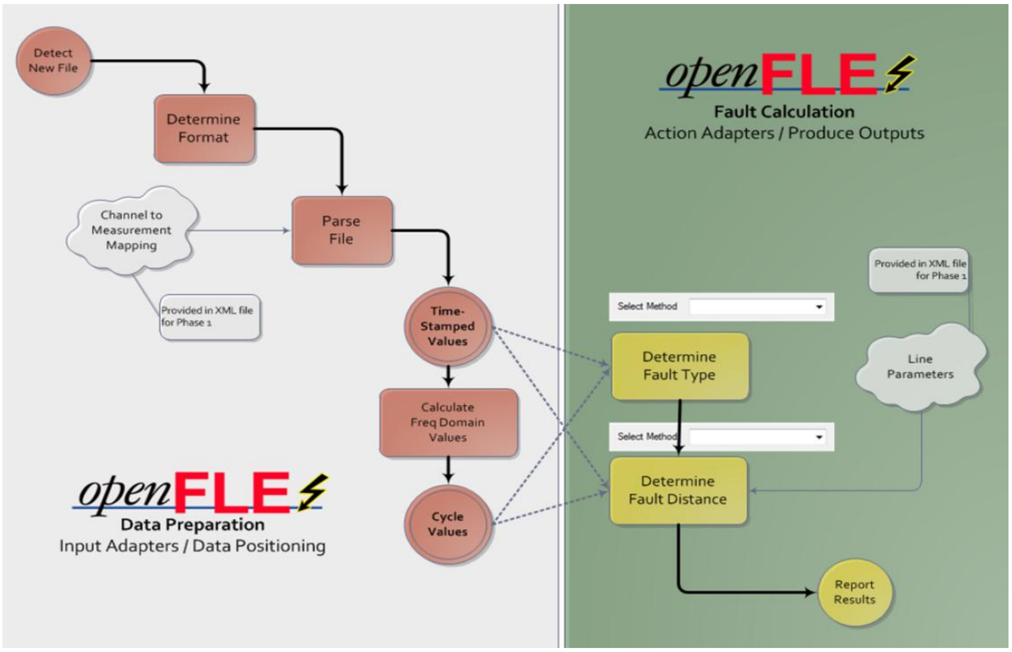

Figure 2 Processing Steps in openFLE

Using 20 test cases derived from actual TVA event records, the demonstration was a success. As soon as the required input conditions were met, the openFLE automatically processed the event records, and produced accurate results. The test results showed some variation between the hand calculated distances, and those automatically calculated by openFLE, but as you can see in Figure 1 below, they demonstrated the feasibility of an automatic fault location system.

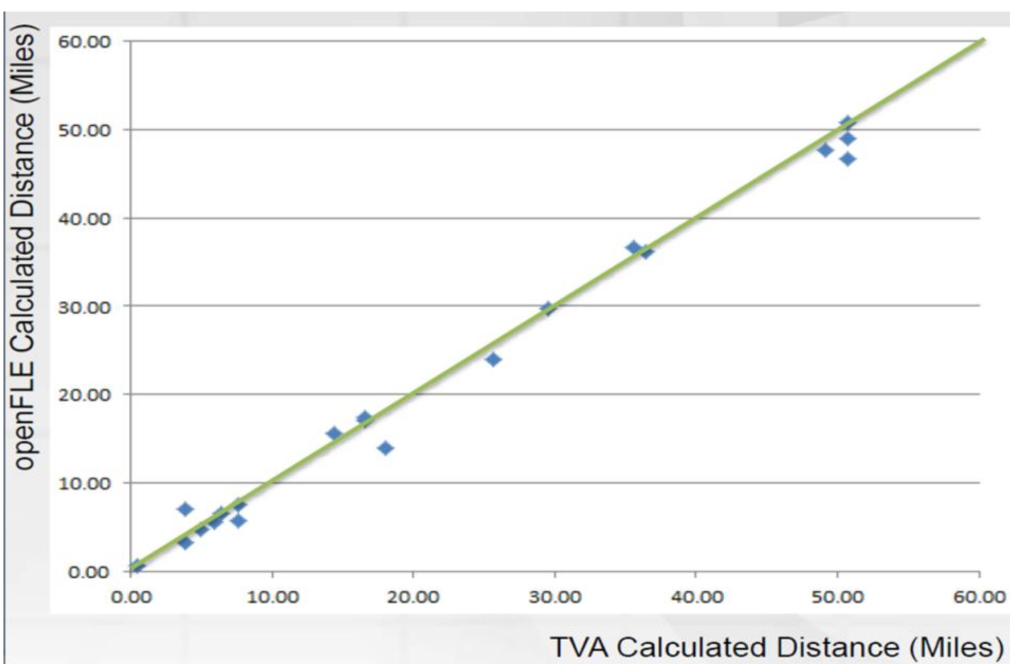

Figure 3 Comparison of Manual vs. Automatic Distance to Fault Calculations

EVOLUTION FROM openFLE to openEDA

After reviewing the success of the openFLE demonstration project, examining TVAs continued desire to have a fully automated system, and carefully considering the power of the platform developed to facilitate openFLE, we realized that this is much more than a fault location engine, and a new name was selected to more accurately describe this larger vision. The new name, Open Enterprise Disturbance Analytics (openEDA), conveys the message that this platform is scalable to create enterprise class systems for disturbance analytics of any kind.

An enterprise wide approach to disturbance analysis and fault location provides many benefits when compared to other approaches. The openEDA platform allows a central location to automatically process every event record as soon as it is received from the remote device. This continuous analysis of every event record assures that information about every fault will be available in the shortest time possible, and systems or individuals can be actively notified of faults. Standard COMTRADE and PQDIF input data formats, removes the need to use multiple proprietary tools, and using a single analysis tool ensures consistent and comparable results. To accomplish the automated processing, it is necessary to have system parameters positioned for the analysis engine to use in combination with the disturbance records. A central enterprise approach simplifies access to other required data sources that could include asset management, right-of-way, and lightning systems, and many others, depending on the scope of the automated system, and the information to be include in the fault report or notification.

With regard to standards, there are a number of things to consider. Input and output is the first one. We chose COMTRADE and PQDIF since they are both used at TVA for event data, and are seemingly "universal". Most IEDs have the option to output one or the other of these standard formats. If there are other standard formats commonly used for event records, it would be reasonable to consider new input adapters to accept them. In our work so far, we have discovered that not all vendors output properly formatted files, even though they claim to be using the standard. This non-compliance to the standard has prevented the use of that data until the file format is correct. The openEDA overview diagram in Figure 4 below, shows SCADA and Phasor Data as possible inputs, and in those cases it would also be required that the data conform to the appropriate standard; C37.118 or IEC 61850-90-5 for phasor data, and DNP3 or 61850 for SCADA. At this time, PMU and SCADA data input adapters are available in the open source libraries, but have not been implemented in the openEDA. Output standards don't apply in quite the same way, since the purpose of the openEDA is to support virtually any downstream application that needs the results of the analytics. Both .XML and SqlServer have been used as output destinations.

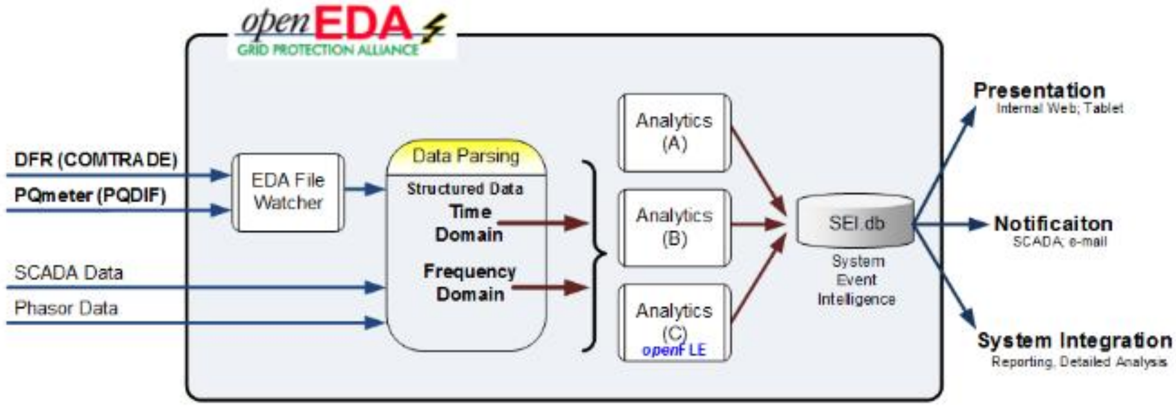

Figure 4 OpenEDA Overview

Another consideration for standards, is the methodology used in the analytics. Since one major purpose of open source is to foster innovation, openEDA does not prescribe a specific methodology. However, the value of the results will be enhanced if the specific methodology used is identified. In our initial openFLE demonstration, simple text book methods were used for fault presence, type, and distance. If specific recognized standard methods are used and identified, their use would set some expectations for the output.

After carefully considering the appropriate use of standards, we need to come back to the realization that the openEDA is intentionally being developed in open source so that it can live and grow according to the community needs. Even though standards are important from a system perspective, unique nonstandard situations can be accommodated without degrading the overall project value. For example, the DFR vendor who does not produce standard COMTRADE files. From our perspective, the 'right' way to solve the problem is for the manufacturer to produce standard files. However, another valid approach would be to write an input adapter that reads their native files. Once that input adapter is in place, all the power of the analytics are available to analyze data from that source the same as any other 'standard' source, and even more importantly, in combination with data from any other source, standard or nonstandard.

MOVING FROM DEMONSTRATION TO THE REAL WORLD

For TVA's pilot project, to implement a completely automated fault location system, ten DFRs are polled once a minute for new records. System parameter files that correspond to each of the lines monitored by the DFRs are positioned for use in the automated analysis, and an event database is used to store the results. A "File Watcher Service" is positioned to start the analysis process as soon as a new event file arrives. See Figure 5 below.

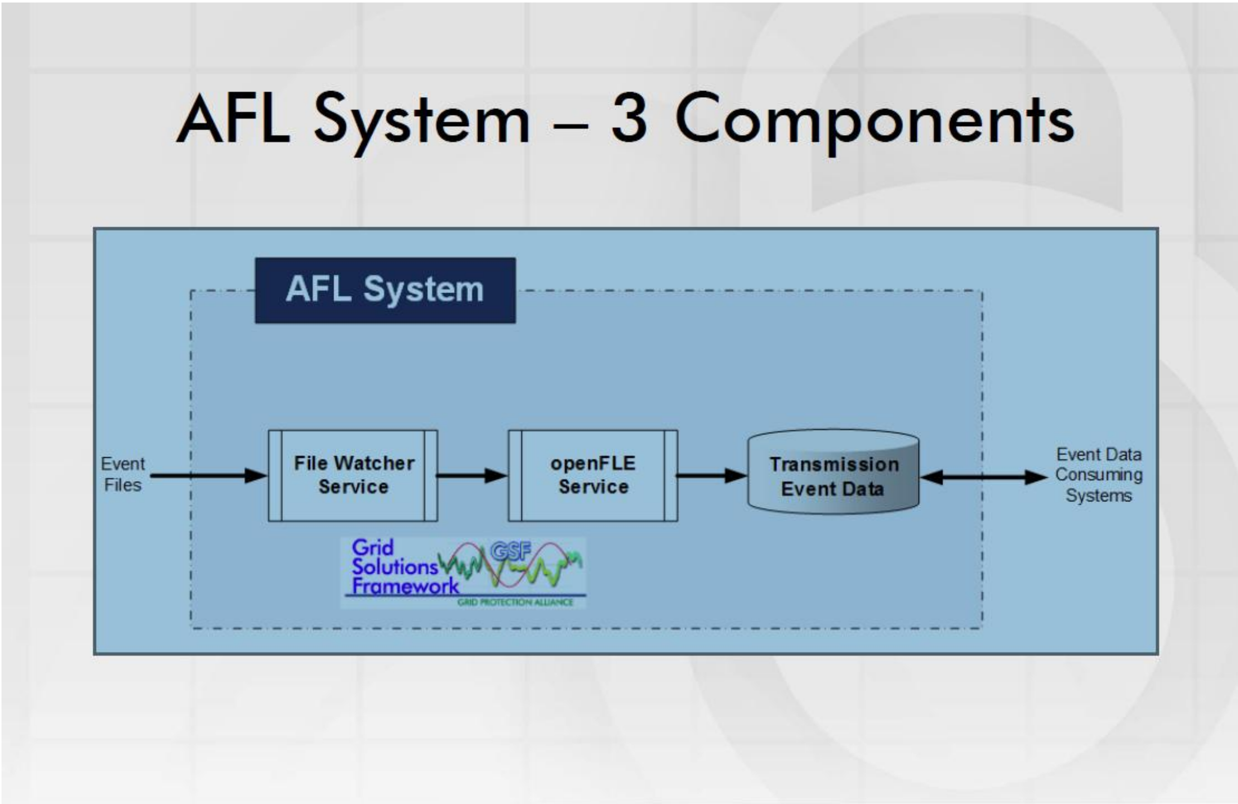

Figure 5 Automated Fault Location Components

At the writing of this paper, all of the components have been developed, tested, and installed at TVA. Initial testing has shown the overall process to work, and final system configuration and population of line parameters is being completed. In developing requirements for this system, the operations manager requested that the fault location data be made available to the operators within two minutes, and all indications are that the system will exceed his expectations. As the system matures, TVA will be positioned to use the flexibility of the openEDA platform to add or change fault analysis algorithms, and consider the addition of other analytics, without the need to develop a new system.

THE RECIPE (CONCLUSIONS)

Ingredients:

1. One or more sources of event records
2. Access to accurately maintained system configuration data[1]
3. An open source fault location engine to perform the calculations
4. An output destination for the results

[1] Please note that once synchrophasor data is available at both ends of every line, it will no longer be necessary to access system configuration data repositories, since impedance values can then be measured in real-time rather than calculated from historical asset data. Complete coverage of the system with synchrophasor data is now possible by enabling PMU data outputs on most IEDs.

Directions:

Combine the ingredients carefully, understanding that the resulting system can be flavored to taste, ranging from simple desktop manual input processes of copying files to a specified input folder, and then manually opening up the resulting output files, all the way to a fully automated system that processes event records as soon as they are retrieved, and actively publishes the results to notification systems. Please note that as you gather the ingredients together, and move toward more automated processes, you are likely to expose opportunities for improvement in your existing systems and processes. However, the finished product can produce satisfying and valuable results, and may in fact allow you to operate your system more effectively. As you bake your very own open source fault location system, please mark up this recipe and share with others so that the open source fault location community can be healthy and growing.