GTC Fault Trace Tool

Authors:
GTC - Marlin Browning, Lori Hartzog, Joey Lowery
Lifescale Analytics - Theo Laughner

Abstract

When a fault occurs on the power system, it is imperative to identify the fault location so that restoration efforts can begin. Previously, GTC implemented an automated fault location system that produced a report with the distance to the fault. However, due to the nature of the GTC network, the calculated fault location could result in multiple physical locations on the system. Consequently, GTC decided to develop a GIS-based fault mapping system called the fault trace tool (FTT). The FTT receives input from the automated fault location system and produces a map with all of the possible fault locations. The FTT also lists the structure ID's nearest to the computed fault distance. This paper shares the features of the system, the benefits from the system, and the lessons learned during the deployment.

Background

Georgia Transmission (GTC) is responsible for providing transmission services between 38-member electric cooperatives and the generation companies in the state of Georgia. Figure 1, below, shows the service area for GTC.

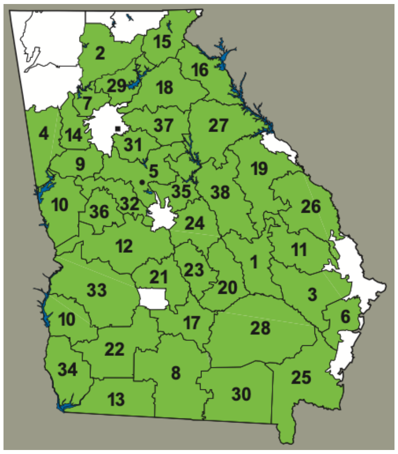


Figure 1 - GTC Service Area

Reliable and affordable energy, innovative technology, and Smart Grid are all key priorities for GTC. In keeping with these priorities, the protection department deployed digital fault recorders across the system. These devices record faults on the system and reports the events to a centralized collection system. The collection system, in turn, performs an automated analysis of the data and calculates a distance to faults on the lines. Investment in these tools have improved the speed with which maintenance crews can respond to faults on the lines. Figure 2, below, shows an overview of the automated fault location system.

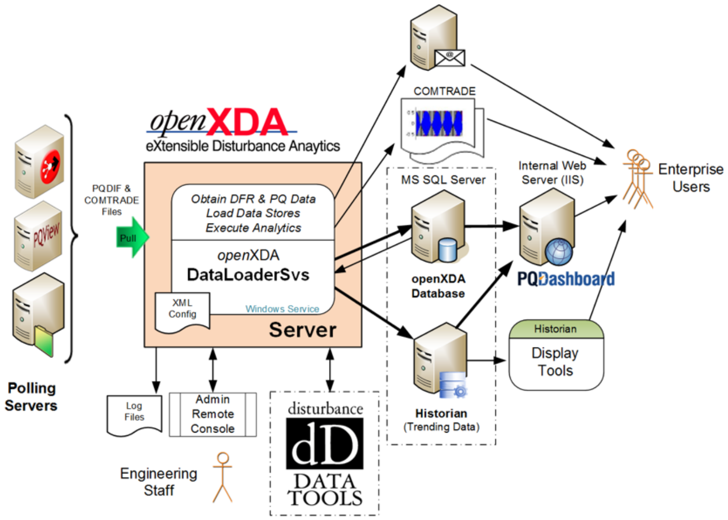


Figure 2 - Overview of GTC Fault Location System

The Challenges

While the system has improved the response time of the maintenance crews, enhancements were developed to provide additional insights into fault locations. The existing tools informed crews about the distance to the fault along a transmission line. However, GTC has many lines which are tapped. Consequently, a single fault distance may result in multiple physical locations. Consider figure 3, below, which shows how a single fault distance could be at multiple locations on a transmission line.

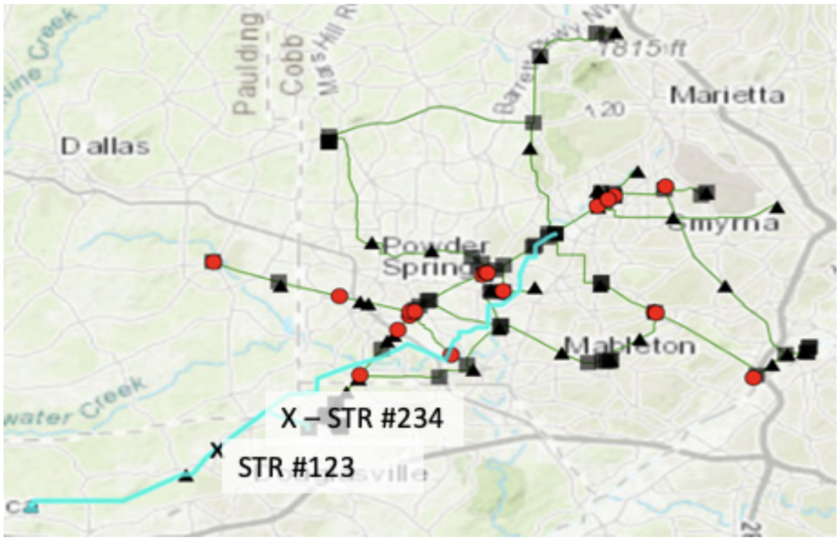


Figure 3 - System with multiple branches

In addition to users being able to manually interact with the tool, GTC wanted to be able to integrate the tool with the existing systems so that automatic fault location emails could be sent with maps included. This required there to be an application programming interface (API).

Coordination meetings between multiple vendors were required to ensure that the systems could talk with each other.

Almost all of the data about the transmission system was in the geographic information system (GIS) at GTC. However, GTC was in the middle of a significant effort to improve the data within the GIS. As a result, some of the data needed to be fixed during the project. In some cases, not all of the data had the correct attributes. In addition, the GIS did not always contain a complete map of the connectivity between the substations and multiple line segments. Consequently, the tool would need to be able to be robust to ensure that changes to the GIS would not break the tool. In addition, the tool would need to be flexible to support the underlying changes in the data.

A high percentage of faults on the GTC system are caused by lightning. As a result, GTC also wanted the ability to show lightning in the line right of way that was time-correlated to the fault. GTC already had a different vendor in place to provide lightning data. Fortunately, the vendor provides their own API to access lightning information through the internet.

Finally, the data in the GIS represented the geographic representation of the data. It did not also contain the network topology or connectivity between the lines, substations, and structures. In some cases, lines were comprised of multiple line segments. Necessarily, the tool needed to be able to develop the topology of the network on the fly keeping in mind that the GIS could change at any time as updates were made.

The Approach

In order to reduce risk, a 3-phase project plan was developed. In phase 1, a basic user interface and web-API were developed to show the locations of faults on a map. In phase 2, lightning was integrated into the tool. The final phase, now in progress, is incorporating the status of fault indicators from the GTC SCADA historian.

Each phase had multiple tasks (design, develop, deploy) to ensure the solution was moving as expected. During the design task, technical and functional specifications were developed to ensure that all stakeholders had a shared vision of the tool. Staff from multiple departments within GTC and from multiple vendors were coordinated through bi-weekly coordination meetings during the projects. Corresponding functional and technical specification documents were developed. The functional requirements articulated the business process and data requirements, while the technical requirements enumerated the hardware and software requirements for each component. Risks were also identified during the development of the specifications. In addition, project success criteria were also developed. Finally, a user acceptance and test document was developed to ensure that the tool worked as expected.

Coordination meetings continued throughout the project to ensure the deliverables from each vendor were on time and budget. These meetings were also used to identify emerging or unforeseen risks and develop a remediation strategy.

At the end of each phase, a user acceptance test was performed. Any issues identified during testing were remediated. Finally, the developed tool was moved into the production environment.

The Solution

Figure 4, below, shows the fault trace tool architecture. Each block is colored based on a vendor/stakeholder touchpoint. As is shown, a number of systems and support staff are needed to be engaged to have a successful project. In phase 1, the FTT was connected to the fault servers and GIS servers. In phase 2, the tool was connected to the lightning server. Finally, in phase 3, the tool was connected to fault indicator data in the SCADA system.

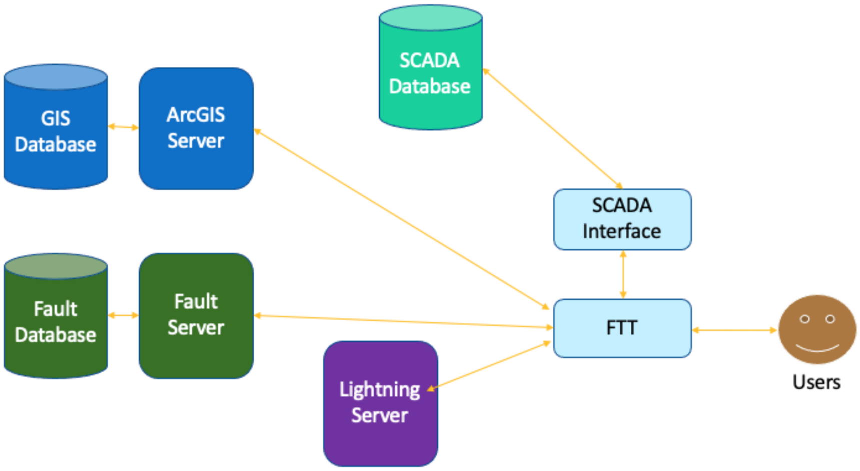
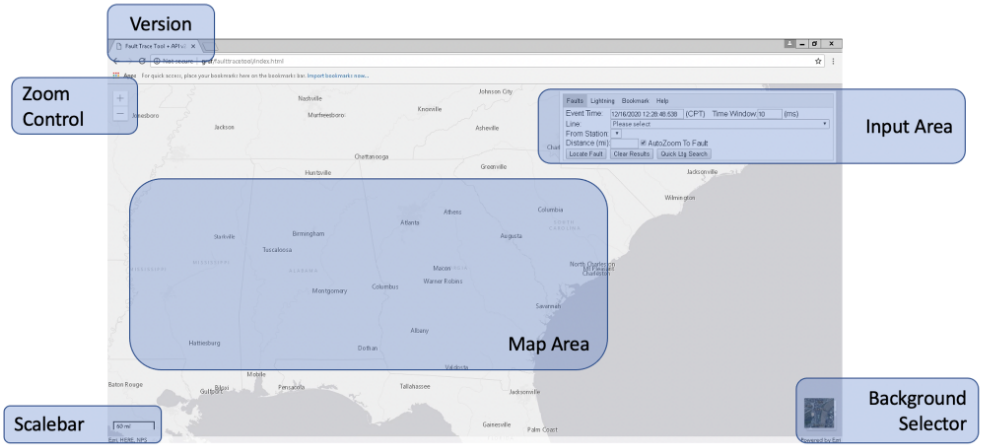


Figure 4 - Supporting Architecture for Fault Trace Tool

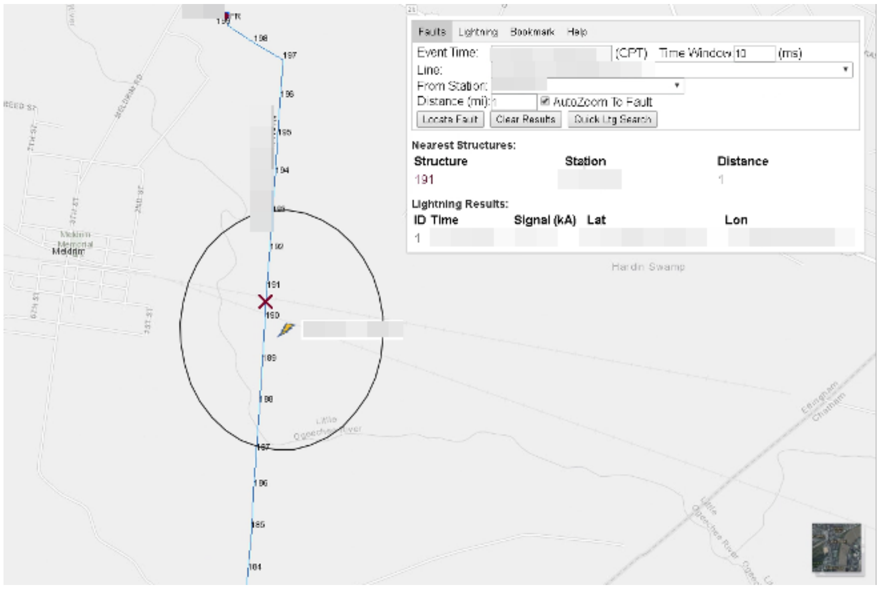
A user interface was developed to enable staff to manually locate faults on the power system. The map-centric user interface is shown in figure 5, below. The user interface has an input area where users can enter the fault time, select a line, select a substation, and enter a distance.



Event Time: 12/16/2020 12:28:48.538 (CPT) Time Window: 10 (ms)
Line: Please select
From Station:
Distance (mi):
☑ AutoZoom To Fault

Figure 5 - Fault Trace Tool User Interface

After clicking the locate fault button, the results appear on the map as shown in figure 6, below. The user interface zooms to the selected line and the structure nearest to the fault. An X is used to denote the location of the fault on the map. In addition, a lightning query is run to see if there is lightning near the line. If lightning is found, then a lightning strike will appear on the map along with information like time stamp, signal strength, and location shown in a table.



Event Time: [redacted] (CPT) Time Window 10 (ms)
Line: [redacted]
From Station: [redacted]
Distance (mi): 1 ☑ AutoZoom To Fault

Nearest Structures:

Structure	Station	Distance
191	[redacted]	1

Lightning Results:

ID	Time	Signal (kA)	Lat	Lon
1	[redacted]	[redacted]	[redacted]	[redacted]

Figure 6 - Example output from fault trace tool (some areas redacted)

In addition, a web API was developed to enable 3rd party systems to generate output from the tool automatically. The API utilizes a standard web address with parameters that enable the systems to specify the station, the line, the distance, and time window to be used for analysis. An example of the API is shown below.
http://<<servername>>/faulttracetool/index.html?station=<stationid>&line=<lineid>&distance=<distance>&eventtime=<datetime>&timewindow=<timewindow>&outputType=jpg

- Station = Station Name
- Line = Line ID
- Distance = Distance in Miles from Station
- Eventtime = Timestamp for event
- TimeWindow = Number of seconds to search around event time.
- zoomToFault = {y = yes, n=no}
- OutputType = { jpg=jpeg, h=html, json=json }

Both the API and user interface are an HTML and JavaScript application built, in part, on ESRI's ArcGIS JavaScript API. This allows for a rich, map-based user experience. The line list is populated from the line features in the GIS. When a line is selected, the substation list is populated by automatically filtering substations physically close the line. In addition, the structures associated with the line are displayed.

When the user clicks the Locate Fault button, the tool builds a topology from the map assets. Next, the topology is used to trace the line from the selected substation to the vertex on the line nearest to the fault distance. Finally, the structure nearest that vertex is selected.

After the fault is located, the tool produces a buffer around the selected line. The buffer geometry and the time window are sent to the lightning system. When the lightning data is returned, it is displayed on the map. Figure 7, below, shows the process diagram for the lightning module. Additionally, Figure 8 shows a sample automated fault email.

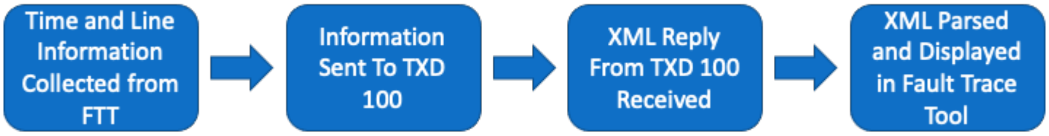
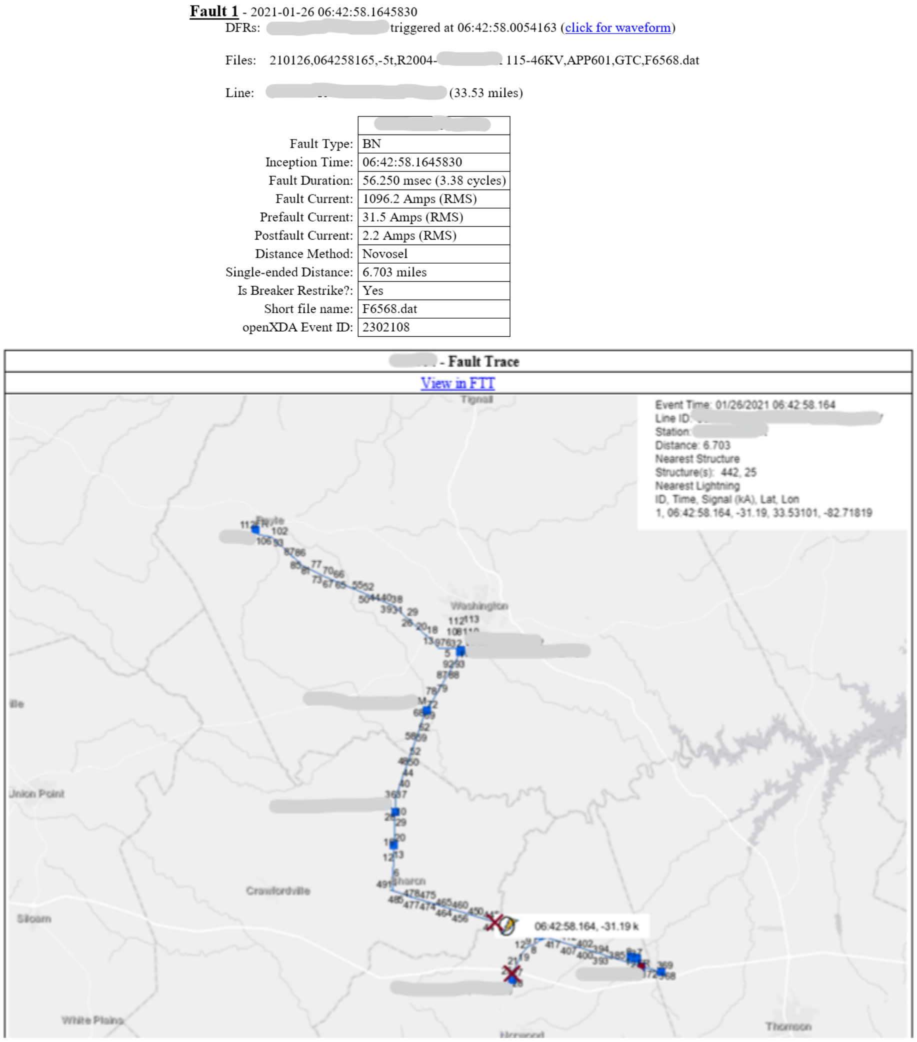

Time and Line Information Collected from FTT
Information Sent To TXD 100
XML Reply From TXD 100 Received
XML Parsed and Displayed in Fault Trace Tool

Figure 7 - Lightning Process Diagram



Fault 1 - 2021-01-26 06:42:58.1645830

DFRs: [redacted] triggered at 06:42:58.0054163 (click for waveform)

Files: 210126,064258165,-5t,R2004-[redacted] 115-46KV,APP601,GTC,F6568.dat

Line: [redacted] (33.53 miles)

	[redacted]
Fault Type:	BN
Inception Time:	06:42:58.1645830
Fault Duration:	56.250 msec (3.38 cycles)
Fault Current:	1096.2 Amps (RMS)
Prefault Current:	31.5 Amps (RMS)
Postfault Current:	2.2 Amps (RMS)
Distance Method:	Novosel
Single-ended Distance:	6.703 miles
Is Breaker Restrike?:	Yes
Short file name:	F6568.dat
openXDA Event ID:	2302108

[redacted] - Fault Trace

View in FTT

Event Time: 01/26/2021 06:42:58.164
Line ID: [redacted]
Station: [redacted]
Distance: 6.703
Nearest Structure
Structure(s): 442, 25
Nearest Lightning
ID, Time, Signal (kA), Lat, Lon
1, 06:42:58.164, -31.19, 33.53101, -82.71819

Figure 8 – Fault Email (some areas redacted)

Results

GTC has been very pleased with the results of the FTT integration. The map provides a quick visual snapshot of fault location(s), lightning in the area, and enables GTC to visually analyze the fault information from our DFRs. The transmission line maintenance team and control center operators now only have to open the FTT map within the OpenXDA fault email to find out more information about the fault, instead of using two systems (OpenXDA fault email and our GIS system). This allows for a quicker response to outages, and the benefit of seeing where the fault is before sending crews out to investigate. For outages not involving fault recorders, our outage investigators can use the manual FTT to run a study on a line with relay fault data information. Additionally, whenever our EMC Members call us with lightning information requests at a specific location, we can input the address into the web FTT to find out lightning within a timeframe of the address. GTC looks forward to utilizing the visual map inclusion of our Fault Indicators on multi-tap lines to narrow down the location of a fault, thus enabling our crews to quickly know exactly where to go.

Lessons Learned

During the project a number of lessons were learned. The GIS Data needed to have some quality control in order for the project to be successful. The Windows server needed to be upgraded to accommodate some of the features. The interface between the automated fault system leveraged a web browser control that needed to be replaced during the project. All of the systems use different time stamp formats and time zones. Finally, station names were not the same in both the automated fault system and the GIS.

There were several lessons learned about the data within the GIS. The lines in the GIS were not snapped to the structures in the GIS. As a result, the actual fault distance and the nearest structure could be at different locations. Connectivity along the line was broken at times which prevented the FTT from walking out the line. In other cases, lines were broken into segments, but the segments may not be in a predictable order making it necessary to compare ends of line segments to ensure that segments were in the correct order. Substations were not always snapped to the line end making it necessary to look for substations within a certain distance of a line. In some cases, this made substations to be on the line appear that were not. Attributes are being added to many of the features to ensure that they are properly correlated in the future.

After the release of phase 1, the server operating system had to be upgraded. During development and testing the operating system of the server was Windows 10 and Windows 2016. However, the production server was Windows 2012R2. The code failed to run on Windows 2012R2. After careful testing and consideration, the decision was made to move the tool to a new Windows 2016 server which resolved the issue.

During the development process, the web browser control in use by the .NET development framework stopped using Internet Explorer and started using Chromium. This caused problems for headless operation in use by the automated fault system which took a considerable amount of development effort to overcome. Ultimately, a new control was found to resolve the issue.

Even though the GTC footprint is in the Eastern time zone, the system is operated in Central Prevailing time which observes daylight savings. The user interface had to make sure that all time zones from external systems were coordinated to central time. The automated fault system uses UTC which does not observe daylight savings. The lightning interface also uses UTC. However, while the SCADA system uses UTC internally, it automatically adjusts the data to the time zone of the computer making the calls, in this case Eastern. Consequently, all time related calls had to be carefully formatted. In addition, each system formatted the date and time in different formats (mm/dd/yyyy vs mm-dd-yyyy vs yyyy-mm-dd). Internally, these times were all normalized to a common format and time zone.

The automated fault system only stores substation names. The GIS has both station names and station IDs. However, the station names did not perfectly match in both systems. The decision was made to leave the existing station names to reduce the need for maintenance. The API for the FTT implemented a fuzzy matching system to align the names between the systems.

Conclusion

The Fault Trace Tool was developed to help GTC respond to system events more quickly. The system has successfully helped GTC identify the fault location and the cause when the fault is lightning. The project was complex due to the number of different vendors that needed to be coordinated. Good project management and planning helped reduce risk and delivered a successful outcome.