Interoperability of System Protection Software

Jared Gurley
Oncor Electric Delivery
Fort Worth, TX USA
jared.gurley@oncor.com

Luci Hays
Oncor Electric Delivery
Fort Worth, TX USA
lucilia.hays@oncor.com

David Ridley
Oncor Electric Delivery
Fort Worth, TX USA
david.ridley@oncor.com

Zachary Austin
Oncor Electric Delivery
Fort Worth, TX USA
zachary.austin@oncor.com

Nathan Thomas
SynchroGrid
College Station, TX USA
nthomas@synchrogrid.com

Luke Hankins
SynchroGrid
College Station, TX USA
lukeh@synchrogrid.com

Alberto Borgnino
SynchroGrid
College Station, TX USA
alberto.borgnino@synchrogrid.com

Joe Perez
SynchroGrid
College Station, TX USA
jperez@synchrogrid.com

***Abstract*—Software is used in system protection for a variety of reasons such as data management, performing short circuit simulations, and automating tedious work, all aiming to empower engineers to continue developing reliable protection schemes for an increasingly complex power grid. We have found that *interoperability* between software packages such as the short circuit model, relay setting database, and automated relay setting calculation software, enables sophisticated, streamlined processes. With seamless interaction between software, data is accurately and efficiently exchanged, engineering resources are less strained, and compliance requirements such as NERC PRC-027-1 [5] R1 and R2 are more easily met.**

We present a case study of Oncor Electric Delivery's software integration efforts, describing both methodology and benefits realized. Oncor utilizes multiple software tools which are composed together to implement standardized, automation assisted workflows for system protection activities. Beyond the standard capabilities of each application, the company realized substantial benefits from these efforts, avoiding the common inefficiencies and errors that exist at the boundaries and interactions of such tools. This more holistic, integrative approach is made possible by next-generation application programming interfaces (APIs) that expose greater access to the capabilities of each software solution. In addition to more efficient data transfer, these APIs allow one program to more directly interact with another, requesting actions be performed (e.g., fault simulations) on its behalf. Novel functionality can thus be created, leveraging each application's strengths to create a single, automated process.

We present a relay settings development process (PRC-027-1 R1) with automated settings calculation software that interacts with the short circuit model, generates reports as well as relay configuration files, and stores results in the database. Second, grid model consistency is ensured by a novel algorithmic approach to relay database and short circuit model linking, enabled by integrative analysis of both assets from the database and grid model topology inspection. We next discuss lessons learned and then describe ongoing work using interoperablity for wide area coordination studies (PRC-027-1 R2). This work is based on automated analysis that empowers engineers to visually resolve issues which are then documented in automatically generated, per relay reports that are stored in the database. We then conclude with future work discussion. In summary, we aim to convey that software integration in system protection is both necessary and beneficial to increase efficiency, reduce errors, and equip utilities to handle the increasing complexity of engineering activities.

***Index Terms*—System Protection, Microprocessor Relays, Relay Settings, Software Interoperability, Process Improvement**

I. Introduction

There is growing consideration in many system protection teams to embrace software automation as a foundational component of relay settings processes. While basic, ad hoc automation (e.g., scripts for data gathering and manipulation) have for some time been used to augment existing processes, some utilities are choosing to take a more *holistic* view of how ongoing software advancements in system protection might inform a change in process.

One emerging trend in system protection software that serves as a catalyst for process innovation is software *interoperability*. For some time, many industry applications possessed some built in automation capabilities, such as scripting language in short circuit models. These are used by some engineers, but their specialized and proprietary nature have limited their use in creating maintainable, enterprise grade solutions. Attempts at *integrated* solutions with other software (i.e., relay databases and relay settings development packages) required tenuously cobbling them together with custom data management scripts that inevitably break with software upgrades and labor turnover.

Increasingly, system protection software vendors are addressing these deficiencies by providing modern application program interfaces (APIs) which allow their execution to be guided and data retrieved by 3rd party applications. This addition dramatically accelerates innovation, as these packages can be easily *composed* with others, creating integrated solutions built with modern programming languages and libraries. We will discuss several integration examples in this paper and discuss the new types of solutions they are enabling.

Oncor Electric Delivery recently completed a software integration effort which was part of a broader initiative to increase the efficiency of the relay settings development process. In addition to existing short circuit software, the company deployed relay database software in 2018 followed by automated relay settings calculation software in 2021. These third party solutions teamed with internal tools are now combined into an end-to-end, automation assisted process.

Together with interoperability, the decision to move forward came in the context of a growing familiarity with these software packages, settings work growth increasingly outpacing that of personnel, and a revisit of the settings process driven by PRC-027-1. The goals of the project were:

- **Error Reduction** - By augmenting engineering expertise with automation throughout the process workflow common sources of errors, such as copy-paste, are simply removed. Furthermore, by enforcing a standard philosophy across the system, errors caused by variable application are eradicated. Finally, tedious tasks such as testing points generation are completely automated.
- **Increased Efficiency** - Assistive automation and error removal reduces the time required to develop, review, and deploy settings.
- **Reduce Engineer Turnover** - By streamlining repetitive tasks such as fault simulations and report generation, a more sustainable process has been established.
- **Streamlined Compliance** - With growing grid complexity and the need to perform frequent, recurring analysis as part of compliance activities, creating automation solutions to streamline these efforts is necessary.
- **Enable Data Analytics** - Whatever is measured can be improved. An integrated workflow that tracks effort at each step sets the stage for continuous process refinement.

The outline of this paper is as follows: In Section II, we outline the new process for relay settings development (PRC-027 Requirement 1) that was developed and then how interoperability plays a role in its realization in Section III. We next discuss in Section IV how improvements to the grid model were enabled by software integration efforts and then discuss lessons learned in Section V. We then briefly discuss work we are doing for wide area coordination in Section VI and then conclude with a discussion of future work.

II. Relay Settings Development Process PRC-027 Requirement 1

One of the most fundamental processes in system protection is new relay settings development. The importance of a settings development process that is both documented and followed is a fundamental aspect of the recent NERC PRC-027 [5] standard. In this section, we provide an overview of Oncor's new process and describe its development from requirements gathering to implementation. While we give some mention to the software tools used here, they and their interoperability are discussed in detail in Section III. We will focus on the relay settings database (RSD), short circuit model (SCM), and relay settings calculation (RSC) software interactions.

A. Process Requirements

Oncor began developing the new relay settings process with a review of the requirements of PRC-027 Requirement 1, to ensure that the new process would be compliant to R1.1 (review and update of short-circuit model data) and R1.2 (review of developed settings). This was follow by a review of the current relay setting process and brainstorming meetings to envision what the new process might look like. Next, new process requirements were developed including the following:

- **Uniform** - used for all relay settings (transmission and distribution).
- **Familiar** - based around how most relay setting engineers create relay settings.
- **Integrated** - utilize the existing relay settings database and controlled by its statuses and security settings.
- **Documented** - used as a reference document by all relay setting engineers.
- **Measured** - extract process data out of the relay settings database to gain a better understanding of the following:
  - Total number times each step was completed.
  - Amount of days spent on each process section.
  - Engineer who is completing each task.
  - Types of errors occurring and counts of each type.
  - Number of times a rework of setting is required.

Fig. 1. Original Process Workflow Statuses

B. Analysis of the Existing Process

The existing process (depicted in Figure 1) called for engineers to first create a new project in *Draft* in the relay settings database (RSD). The engineer would then progress the status of the project to *Review* for a final review, and *Issued to Field*, with each of these transitions happening when a phase was complete. This linear progression was not in practice, however, how the group integrated the process into their workflow. Instead, engineers would start a project and send deliverables to an experienced engineer for review before ever starting a new project in the RSD. Once the review had been completed and all deliverables had been developed outside the purview of the RSD, the engineer would start a new project in draft, upload all their documentation, and mark it for review. Because the review was completed independent of the RSD workflow, the *Draft* and *Review* statuses in the database were logged with times not correlating to the actual time spent on the project. The resulting data metrics' inaccuracy about the required effort from the setting engineer and reviewing engineer made analysis for process improvement difficult and led to the the *Measured* requirement of the new process.

Once the project had been completed and reviewed, configuration files were distributed to the field technicians. The RSD contained a method for the hand off of documentation and configuration files from engineers to protection and control technicians. Technicians would receive an email when new work was released. The technicians would then download the documentation and configuration files from the RSD to a companion testing software. This synergy provided a means of linking and documenting the test results, any applicable compliance related information, the originally issued settings, and as left settings, that may include more device specific information in the configuration file not available to the engineer at the time of coordination. While there were clearly problems in this process with regards to logging the time it takes to complete engineering projects, the database provided the relay settings team greater visibility into what projects were performed by each engineer, what the status of the relay settings were for those projects, and provided validation of the configuration files applied to the relays in the field. Before the adoption of the RSD, receiving *as left* configuration files was far less common.

C. Design of the New Process

Oncor's definition of new process requirements brought clarity as to what the process needed to look like. Designing it began with drafting in a diagramming application which allowed rapid, iterative refinement as development progressed and eventually yielded a more detailed flowchart of the process. The detailed form was necessary to decompose higher level steps into either decision steps or side steps. Next, a process document was developed that matched the steps of the flowchart but provided in depth descriptions of each step, references, and definitions. After both the flowchart and process document were created, they were analyzed against every type of settings Oncor calculates to ensure that all needed items were included.

Oncor also developed an emergency process which circumvents the standard process when speed is of the essence. For *familiarity*, the emergency process needed to be similar to the normal process yet provide engineers the required flexibility for emergency situations. A primary goal of this process was to reduce the time spent early in the process so that settings can be released to field technicians quickly. We observed that the review steps required the most amount of time to complete, apart from the actual relay setting calculations. Hence, in order to both meet the review requirements of PRC-027 R1.2 and reduce time to release to field, the review step was shifted to after the field technicians had completed work. While this achieves the desired goal, it does result in more manual steps and higher total time to complete the process.

D. Scoping and Analysis of Process Controls

Because the relay settings engineering group was already using the relay settings database, the clear path forward was to implement process controls there, leveraging its rigid status controls and the persistent, automatic storage in the database structure. This approach provided many gains over other methods considered, including minimization of training due to existing workflow and software familiarity.

1) Process Controls Development: During the new process development, the limitations and existing structure of the relay settings database were evaluated. Pragmatic consideration of how engineers would interact with the process also greatly influenced the design of process controls. These efforts ensured the final solution was not infeasible to implement in software nor overly burdensome to engineers and other users.

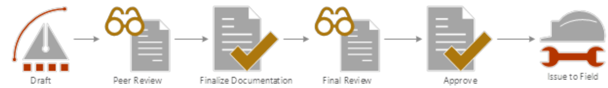

Fig. 2. New Process Workflow Statuses

TABLE I
PROCESS STATUS COMPARISON

Status Number	Old Process Status	New Process Status
1	Draft	Draft
2	Review	Peer Review
3	Approved	Finalize Documentation
4	Issued to Field	Final Review
5		Approved
6		Issued to Field

Controls are assigned between two different engineering security groups, *relay setter* and *relay setter reviewer*, and these groups dictate what steps a user is able to approve. All engineers are able to create a *Draft* project and move the status of the project to *Peer Review*, *Finalize Documentation*, and *Final Review*. *Approve* is limited to relay setter reviewers who are experienced and perform the final review before the original engineer issues the documentation and configuration files to the technicians. The relay settings database captures which user progressed the project to each subsequent status. As an example, consider three engineers:

- **Sarah** - Lead engineer on a project at Station A.
- **Phillip** - Peer review engineer.
- **Bennet** - Region lead engineer and PE for final review.

When Sarah begins her project at Station A, she creates a new *Draft* in the relay settings database. Once her coordination and documentation are complete, she moves the status to *Peer Review* and Phillip is notified that her work is ready for review. Phillip reviews her work and if he agrees, will progress the project to *Finalize Documentation*. If Phillip has corrections for Sarah, he provides her with notes and reverts the status to *Draft*, where the process restarts. Once the project is in *Finalize Documentation*, Sarah will complete her documentation and configuration files and progress the status to *Final Review* which is performed by Bennet. If Bennet agrees with the documentation and coordination, he will progress the status to *Approved*. If Bennet has documentation corrections for Sarah, he will revert the status to *Finalize Documentation* and if he has any major coordination corrections he will revert the status to *Draft* where Sarah will start the process again. Figure II shows the name captured by the relay settings database as the project moves through its lifecycle.

The workflow provides a log of each engineer's work. As all data is in the database, metrics are generated to gauge the development of engineers, as their revision counts drop over time. By logging the user, time, and date of the progression, we can perform further analysis to quantify the average amount

TABLE II
STATUS NAME LOGGING IN THE DATABASE

Status Name	DB Log
Draft	Sarah
Peer Review	Sarah
Finalize Documentation	Phillip
Final Review	Sarah
Approved	Bennet
Issued to Field	Sarah

of time spent per project type per step per engineer. This provides greater insight into the distribution of projects across the engineering group and across the year. These insights enable the optimization of work distribution across engineers.

With the addition of *Peer Review*, we increase reviewer diversity. Previously, only experienced engineers reviewed projects. Now, new engineers review documentation and coordination from more experienced engineers, gaining experience, preventing isolation amongst the engineering group, and promoting the sharing of best practices.

2) Implementing and Testing New Process Controls: Transitioning to the new process began in a pre-production environment for testing. This pre-production environment was a copy of the production environment to ensure the process changes would work with the data it would eventually be deployed on. We began by converting the names of statuses as detailed above. Because of the limitations of the way our previous process was created, some historical status values needed to be migrated. For example, Status 3 was previously *Approved* and was now *Finalize Documentation*, and therefore the status number was moved to 5 to maintain consistency (see Table I).

After status migrations, security group permissions were applied and validated. For the emergency process, a check box was implemented on each item in the database, and an additional status named *Emergency Issue* was added to allow it to initially skip the review steps. The check box also serves as an indication on a relay setting that needs to be followed to ensure that the emergency process gets completed. Testing took several months and ensured that the process flowed logically as well as achieved the requirements set out from our analysis of PRC-027.

E. Deployment of the New Process

Once the new process testing was complete, a process change date was chosen and engineers were advised to complete projects using the current process, to avoid delays providing settings to field technicians. During the production outage, the new process was placed into production in a similar manner to that of the pre-production environment. Engineers were trained on the new process the next day. The final result was a well-documented, generalized normal process that met the requirements listed above as well as an emergency process that provides the flexibility to resolve emergency issues.

To satisfy the *integrated* process requirement, the workflow was designed to limit how often engineers would jump between applications and locations to find standard information and perform engineering work. An additional process requirement, *measurable*, is also enabled by this work, with the end goal for all project work from initial project creation to final issuing to be saved and tracked in the relay settings database. Next in Section III, we will describe how software interoperability plays a significant role to meet these two requirements, as well as others such as *uniform* and *documented*.

III. SOFTWARE TOOLS AND INTEROPERABILITY FOR PRC-027 REQUIREMENT 1 PROCESS

We now describe the software tools used to implement Oncor's new process, specifically focusing on the ways in which they interact with each other for end-to-end process efficiency. There are three external tools Oncor employs together with an internal spreadsheet tool that are composed into a cohesive, integrated solution for process automation. Below are the three vendor tools Oncor chose for this process:

- **Short Circuit Model (SCM)** - ASPEN OneLiner V15 [6]
- **Relay Settings Database (RSD)** - PowerBase V7 [11]
- **Relay Settings Calculator (RSC)** - SARA V3 [13]

While these offerings were chosen both because they met functional requirements and the vendors' ongoing collaborations to promote interoperability, the principles discussed in this paper are general and should be achievable with other system protection software exhibiting similar features and modern application programming interfaces.

In Figure 3, we present a simplified overview of the software workflow both the external and internal software tools employ to implement Oncor's new settings process. Briefly, the process begins with a *Draft* project created in the database, from which existing settings have been placed in the model as described in Section IV. Next, the settings calculator retrieves a specification of the protection philosophy in the form of a template from the database and interacts with the short circuit model to create settings and populate the spreadsheet tool. The spreadsheet is then used together with the settings calculator's review module to perform the settings review. All artifacts created during the development process are then stored in the settings database and issued to the field. We describe software interactions in detail in the following subsections and then describe further process refinements uncovered during the integration project that leveraged the synergy made possible by the software packages' composition.

A. Short Circuit Model and Relay Settings Calculator Interoperability

The fundamental functionalities of a relay settings calculator used to implement the process described in this paper are:

- Formally define a utility's protection philosophy and compliance checks in a customizable manner.
- Automatically interact with the short circuit model to gather information such as:
  - **Grid Topology** - Source lines, remote lines, multi-terminal configuration, tap buses and lines, etc.
  - **Grid Characteristics** - Impedances, line ratings, etc.

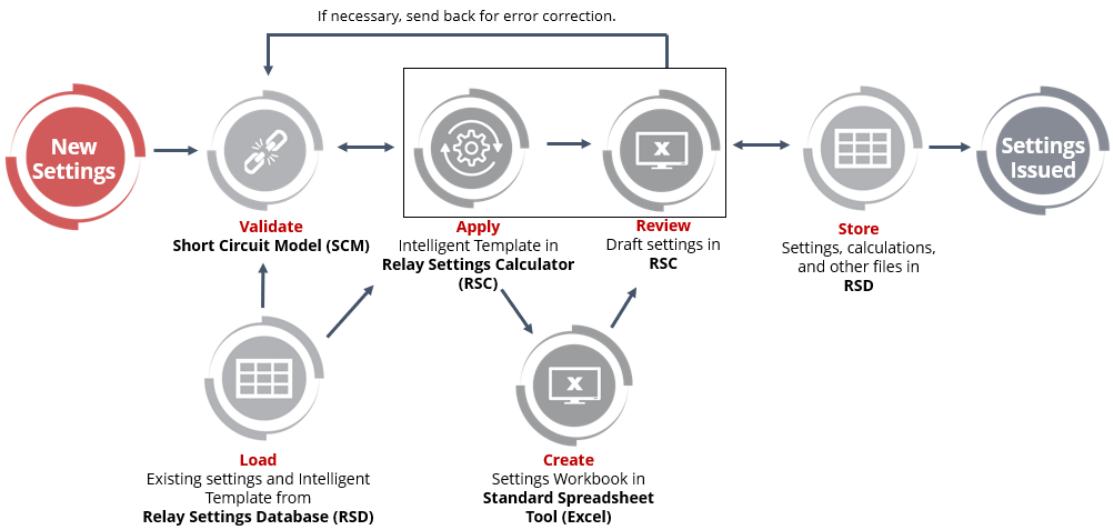

Fig. 3. Simplified Software Process Workflow

– **Fault Derived Calculations** - currents, impedances, relay operation points, contingency analysis, etc.

- Provide an interface to evaluate calculated settings, adjust as necessary, and document justifications.
- Automate ancillary activities such as testing points to further reduce engineering effort.
- Streamline reporting and compliance documentation.
- Generate relay files to avoid copy and paste style errors.

We have previously described our approach to implementing the settings calculator (RSC) and its interactions with the short circuit model in [1]. Figure 4 summarizes the packages' interactions. In this section, we describe recent advancements made possible by a newer API to the short circuit model (SCM).

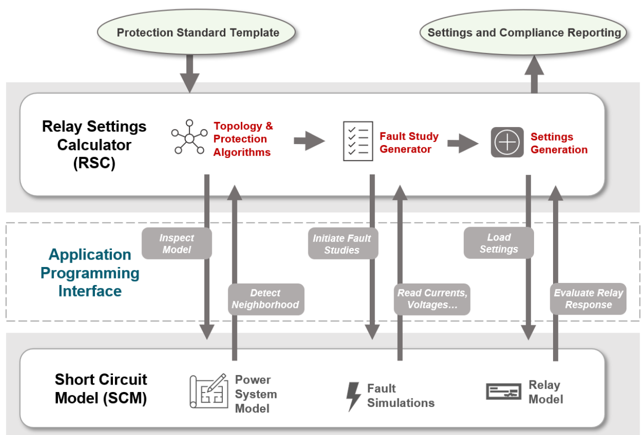

Fig. 4. Relay Settings Calculator and Short Circuit Model Integration.

The latest iteration of the interface between the RSC and SCM is written in modern C++20 [7], an ISO standard which is one of the most prevalent, high performance, and scalable programming languages for systems design. Using this environment allows us to interact with both programs seamlessly and leverage tested and maintained libraries such as the Boost Graph Library [8] we use to represent power grid.

Complex, multi-bus substation analysis, generalized multi-terminal application support, and contingency analysis such as strongest source detection are all implemented using generic graph algorithms [9], drawing from the fundamentals of computer science. Were it not for software interoperability, such sophisticated solutions would not be feasible. Maintainability is assured by using documented, stable APIs between the programs that allow them to safely interact with each other, creating novel functionality to automate settings development.

One example of new functionality built during this project is testing points creation using a customizable template. The RSC dynamically calculates the operation point (i.e., reach) of all distance and overcurrent elements and the corresponding characteristic impedances are used in symbolic mathematical expressions to create a testing point for a given element. For example one might choose to specify the testing point for a Zone 2 distance element to be $(Op_Z(Zone1) + Op_Z(Zone2))/2$, representing a point halfway between Zone 1 and Zone 2 reaches. Using this specification, testing points will then be generated for every line relay application. This expressive capability can be adjusted as the process changes in the future, and in Section III-D2 we describe how the settings calculator presents the information to the settings engineer for review.

B. Relay Settings Database and Relay Settings Calculator Interoperability

In addition to the foundational capability of storing relay settings and other grid asset data, some relay databases provide the capability to specify workflows, which as discussed earlier, allow Oncor to define the steps of the process and trigger automated activities as work progresses. In these integration efforts, we leveraged interoperability between the relay database to remove the manual transfer of settings related data.

For the settings development process, there are two points of interaction. First, philosophy templates are *retrieved* from the relay settings row in the database and used as the protection standard for settings calculation. Second, when calculations are complete, all digital artifacts of the process are efficiently *uploaded* to the database and can trigger subsequent actions such as the review step of the process. These artifacts include:

- Settings calculation sheets (discussed in next section) populated with settings data.
- Relay configuration files (rdbs) with developed settings.
- Relay Settings Calculator software native files

While this is a relatively straighforward use of the database's new API, this integration reduces another potential source of errors and increases efficiency by removing several manual steps previously required by the settings engineer. There are additional interoperability gains between the relay database and settings calculator software for model maintenance. We discuss those in Section IV-B.

C. Relay Settings Database and Short Circuit Model

In order to ensure that relays are properly modeled during fault simulations in the short circuit model, it is imperative that the latest settings from the relay database are loaded there. Discussion of this critical interoperability component is discussed in Section IV. There we find a sophisticated multi-package integration involving all three vendor solutions that proves effective not only in creating a maintainable model, but also discovering some model discrepancies.

D. MS Excel Settings Tool and Interoperability with Relay Settings Calculation Software

During process development, Oncor determined it was an ideal time to change from a document standards (MS Word) to spreadsheet standards (MS Excel), both to save time and reduce the potential for errors. These two benefits come from automatic calculation, interoperability, and the flexibility of a built in scripting language VBA. In this section, we describe how interoperability between the Excel workbook tool and the relay settings calculator software meets the process requirement of *familiar* while meeting overall goals of *error reduction*, *increased efficiency*, and *streamlined compliance*. We first provide an overview of the tool and then explain interoperability with the RSC.

1) Excel Tool Overview: The Oncor excel workbook standards consist of worksheets that each perform a different step of the transmission line relay setting process. On the initial worksheet *Fill In*, the settings engineer inputs the majority of data that will then be propagated where needed through the standardized workflow. This data is both project specific, manually entered data as well as calculated values that relay setting calculator can inject into the worksheet. For engineers or contractors not using an automated calculator yet, these calculations can be performed and entered manually, helping achieve the *uniform* process requirement.

The second worksheet *Questions* presents questions with dropdowns for answers, acting as a logic calculator that compares answers against a lookup table to make appropriate setting changes, such as advanced reclosing logic equations or circuit breaker classification. Though the questions are quite comprehensive, some settings require further analysis. A *Verify* worksheet is presented next that contains a list of settings that the setter will manually verify with a single click.

The last two worksheets in the workflow are the *Front Page* and *My Settings*. They are an accumulation of the work performed on previous sheets which is presented in an easy to read format for documentation and will be referenced to create both a relay download file and setting sheet for the settings. In the past, creating these documents represented a large portion of the settings engineer's time on a project.

The workbook development focused heavily on maintainability for future use within Oncor, with the goal of making it simple for engineers revising the standard to make additions. Furthermore, by integrating manual settings development and automated approaches into a unified workflow, the broader adoption of new automation technology can occur without unnecessary modifications to the process.

2) Interoperability with Settings Development Software: While the Excel tool provides sufficient flexibility to create a settings development workflow that is *familiar* and *uniform*, automating the calculation of core protection settings requires a level algorithmic expertise and interaction with the short circuit model that is only achievable in the relay settings calculator. These capabilities include complex power system typology analysis, a flexible fault simulation engine, and precise tracking of deviations from the standard philosophy.

The Excel tool and the settings calculator are interoperable, as the RSC can both read from and write to Excel worksheets. After automating short circuit model interactions to generate settings, the RSC injects worksheets into the standard spreadsheet which include settings, underlying calculations, engineer notes, and a summary of settings changed from the standard with justification text. Additional information such as mho graphs and dynamically computed testing points (both in human readable and testing set formats) are also generated by the RSC and placed in the Excel tool. These actions, coupled with seamless uploading to the relay database, remove a significant source of copy/paste errors and labor by the settings engineer.

Once draft settings are prepared, reviewing activities are streamlined by a custom review module in the settings calculator developed during this integration effort. The review module provides a framework for the review engineer to manually verify critical calculations, providing an essential layer of check to the automation assisted settings development process. The module's review process is guided by a customizable *review template* which provides the flexible for Oncor to maintain it in the presence of future process changes. Importantly, the module can read as input both the RSC's native file format as well as the Oncor spreadsheet standard. Finally, it can both read input from and write the results of the review directly to the relay database, further streamlining the overall process.

The interaction of the utility's internally created Excel settings tool and a 3rd party relay settings calculation solution demonstrates the importance of interoperability in the adoption of new technologies. The seamless transfer of data between them both allows *familiar* tools to leverage the sophisticated analysis and tight integration the RSC shares with the relay settings database and short circuit model. This transfer ensures that errors are not introduced into the process and creates a synergy in their composition that results in significant automation gains to the overall settings development process.

E. Additional Benefits

Further synergies were quickly discovered during the process development and software interoperability design phases of the project. These benefits include streamlined NERC PRC-023 compliance, improvements in providing relay loadability data to our facility rating teams, and better data analytics. We next describe each of these benefits briefly below and expect further such ideas to germinate with further experience using the new, integrated process.

1) Streamlined PRC-023: During API development between the relay settings calculation software and the relay settings database, we innovated an improvement to the NERC PCR-023 compliance documentation process. This stemmed from the RSC's ability to calculate and directly transfer data that is pertinent to comply with PRC-023 into customizable locations (e.g., asset nameplate fields) in the database. A data location was thus built in the relay setting database that is automatically populated during every setting development project, in a manner specified in the setting calculator software's template for Oncor's protection philosophy.

With this small addition, the latest PRC-023 data will be populated from the relay setting development software to the relay setting database every time that an engineer calculates a relay setting with the relay setting development software. In the end, this reduces the potential for human error and saves substantial time for the protection engineer tasked with maintaining PRC-023 compliance. They no longer manually enter data into a spreadsheet but instead simply review and validate that the data in the settings database is correct. From the database, aggregate reports can be generated to export the data and help prove compliance with this NERC standard.

2) Loadability Limit Calculations: The next benefit from the RSD/RSC API was the automated calculation and database storage of the relay loadability limit. This information is shared with the facility ratings teams to ensure that the relay's load limit is taken into consideration for each terminal across the entire transmission system. This was accomplished performing similar steps as for PRC-023. However, the rating limit can now be automatically included in an email sent to the facility rating teams when the relay settings are *issued* from the relay setting database and have it. This again reduces time spent by the engineer in both calculations and transmitting data to the appropriate staff.

3) Data Analytics: After building the relay setting process into the relay setting database and running our first report on the data, it became evident that many insights could be mined about the setting group's work using data analytics. A large point of interest was quantifying the amount of work being performed and by whom. We were also able to gain insights around how work is distributed throughout the year. Finally, we found that engineers could use the comment field on database settings rows to track errors or other issues found, allowing us to gain understanding around what causes errors to be made and to better understand the issues engineers face. We discuss some of the findings from these analytics further in Section V-A and expect future analysis of such data to help drive further process efficiency gains.

IV. MODEL IMPROVEMENTS

As Oncor territory grows and more stations are built, it is increasingly important to maintain an accurate and reliable system model. The model currently consists of over ten thousand relay elements at approximately two thousand locations. Maintaining relay settings accuracy with the continual growth and change is simply infeasible to attempt by manual means.

The chosen short circuit model and relay database have an existing interface to allow the relay *linking*, allowing settings stored in the database to be seamlessly transferred to relay elements in the short circuit model. Once such links are created, a *bulk import* command allows a rapid refresh of all relay settings in the model. During this project, we created relay linkings for all transmission relays in the model. The relay linking process requires two activities:

- **Settings Translation Scripting** - The short circuit model provides a scripting language to define how externally stored relay settings can be converted into a relay in the short circuit model.
- **Relay Mapping** - Each relay in the relay settings database must be mapped to the correct terminal in the model and appropriate relay elements created for later settings population by the bulk import.

The effort required for translation scripting is notable but inherently scalable, as effort is roughly proportional to the number of hardware relay *models* present in the system. This work is described next in Section IV-A.

In contrast, relay mapping for existing relays was at a scale that Oncor deemed the effort too great a burden to distribute among its engineering staff. Indeed, the magnitude of the task is much greater than translation scripting as it must be done for *every transmission relay in the entire power system*. This is a common problem faced by large utilities that must be addressed so that the model accuracy requirements of PRC-027 R1-1 can be met with a reasonable amount of engineering resources. While new and changed relays in the future can be managed by *incremental model maintenance* defined in the new settings development process, the challenge of relays already in service required a new solution.

We address the relay mapping challenge with a novel automated approach that leverages the intersection of data made available by interoperability between the relay settings database, short circuit model, and settings calculator. This algorithm is described in Section IV-B followed by subsequent verification efforts in Section IV-C.

A. Settings Translation Scripting

During the project, we developed settings translation scripts that are tailored both to Oncor's relay setting standards and the hardware relay models used by the company. These custom scripts were designed to enable the import of relay settings without any engineer input, a necessary requirement for bulk automation. Even used in an ad-hoc manner without relay mapping, the scripts allowed accurate modeling to be achieved much faster than previously possible.

Validation of the translation scripting occurred by having engineers manually map relays for new projects for a period of time and verify the results of the complete relay linking process for the small number of relays required. Verification included not just translated relay settings but also that the correct hardware relay model had been detected. After sufficient testing for correctness as well as estimating the expected reduction of effort for bulk linking of the entire grid, work began on an automated means of relay mapping.

B. Automated Linking Algorithm

Given the challenges described above, we developed a novel, automated approach to relay mapping. Briefly, our mapping algorithm proceeds in the following steps which are also depicted in Figure 5:

1) Given a list of relays to map, extract unique database *locations* to map to short circuit model terminals.
2) Based on utility specific naming conventions and short circuit model information, use a collection of matching techniques successively applied to attempt database to model mappings. Example techniques include:
   - *Regular expression transformations* [10] such as vowel removal or other shortening techniques used by engineers to meet varying length restrictions of the relay database or short circuit model.
   - *Approximate string matching* [14] to account for typing errors or other naming anomalies. For this work, we used a modified Levenshtein Distance algorithm [12].
   - Short circuit model *topology analysis* present in the relay settings calculator. For example, Oncor's relay database naming convention used breaker numbers which were correlated to terminals via breaker associations in the model.
   - *Automated recalculation of relay settings* such as primary line impedances and zone reaches using the settings calculator and short circuit model that are then compared to candidate relay's settings stored in the relay database.
3) For locations matched above, assign confidence scores based on technique that was used. Taking the Levenchstein algorithm for example, the score was reduced for every incorrect character in approximate name matching.

Unlike most of the interoperability described in this paper, the implementation of this algorithm required tight, simultaneous integration of all three system protection software applications, as the algorithm relies on the combined capabilities of the short circuit model, relay settings calculator, and relay settings database. The ability to automatically draw on data stored or computed in all of them allowed us to quickly create a solution which also used existing libraries for regular expression transformations and approximate string matching.

Our program implementing this algorithm generates a spreadsheet summarizing mappings and associated confidence scores. The spreadsheet guided verification efforts and provided a place to record comments for use by the team during the process. Once the mappings were complete, relays were created using the APIs of the short circuit model.

While the algorithm was not able to match all cases, we achieved an overall 91.6% success rate with this initial implementation in the project (see Table III), dramatically reducing the level of manual effort required by the team by an order of magnitude.

C. Verification of Relay Linking

In order to verify the relay mapping and overall linking effort, we divided the system up by work center for review and corrections as needed. For one of the work centers, we exhaustively verified that every relay was matched to the correct item in the settings database. We did this by manually checking the database item that each relay was linked to and verifying that the link pointed to the appropriate in-service settings. Any relays that were mismatched or failed to match we linked by hand. This procedure allowed us to identify some corrections needed in the model and settings database such as identifying stations that were not created in the relay database at the time, name differences as stations were upgraded, or voltage levels that had changed. We also identified a number of setting files missing from the database. This check demonstrated that the algorithm could correctly identify the appropriate mapping in most cases as well as provide accurate suggestions for list of relays to verify or link manually.

We ran the linking algorithm on the remaining work centers, continuing to make any corrections needed by hand. The whole process took about a month (interleaved with other, ongoing work), yielding a verified model with a consistent relay naming convention and all transmission relays linked to the in service settings from the settings database. Table III demonstrates the performance of the linking algorithm by showing the number of relays that were linked automatically and the number of relays that were linked by hand for each work center. It also shows the unexpected benefit of the algorithm's results to identify model discrepancies, that once resolved further increased the accuracy of both the short circuit model and data in the relay settings database.

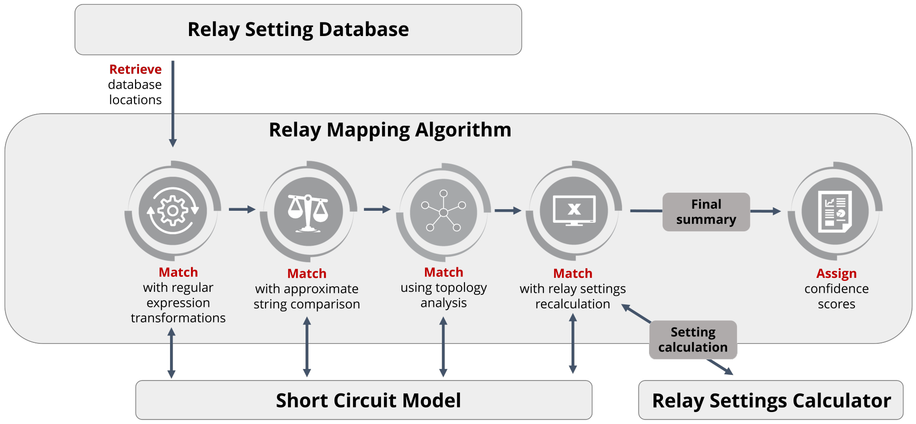

Fig. 5. Automated Linking Algorithm Overview.

TABLE III
LINK ALGORITHM ANALYSIS

Work Center	Relays linked automatically	Relays that needed verification due to model discrepancies	Relays verified or linked by hand	Total number of relays	Percent linked automatically	Percent linked automatically (adjusted for model discrepancies)
#1	253	112	20	385	65.7%	94.8%
#2	88	45	5	138	63.8%	96.4%
#3	102	28	19	149	68.5%	87.2%
#4	81	98	21	200	40.5%	89.5%
#5	299	258	29	586	51.0%	95.1%
#6	121	62	11	194	62.4%	94.3%
#7	249	88	26	363	68.6%	92.8%
#8	99	126	3	228	43.4%	98.7%
#9	146	47	5	198	73.7%	97.5%
#10	46	70	17	133	34.6%	87.2%
#11	51	71	4	126	40.5%	96.8%
#12	39	42	7	88	44.3%	92.0%
#13	112	60	17	189	59.3%	91.0%
#14	76	18	8	102	74.5%	92.2%
#15	32	22	9	63	50.8%	85.7%
#16	36	36	15	87	41.4%	82.8%
#17	59	45	12	116	50.9%	89.7%
#18	40	55	24	119	33.6%	79.8%
#19	46	30	2	78	59.0%	97.4%
#20	31	30	6	67	46.3%	91.0%
System Total	2006	1343	260	3609	53.6%	91.6%

After this effort, the relay settings development process is significantly streamlined, as a settings engineer no longer needs to have the database and the model opened at the same time nor manually ensure data consistency. Settings are now retrieved with a button click within the model, reducing engineers' time spent updating the model prior to creating new settings.

V. Lessons Learned and Challenges

When accounting for initial deployment of software solutions, development of the new process, and the integration efforts described in this paper, this was a multi-year effort involving numerous engineers. In this section, we describe a few lessons learned and challenges faced.

A. Data Analytics

We learned through these efforts that engineers spent significant time performing tasks that could automated. This has led to more efficiency for the relay setting engineers already and even more gain is expected. Depicted in Figure 6 are the total counts for each status of the process prior to switching to the new process. Figure 7 shows the counts for each status after the new process was implemented. It can be determined from these graphs that the new process clearly identifies the work being performed in each status and depicts the actual amount of work being performed. The efficiency gains of the new process can be seen by the increase in overall counts of the statuses from the relay setting database.

A challenge prior to the new process was trying to under-

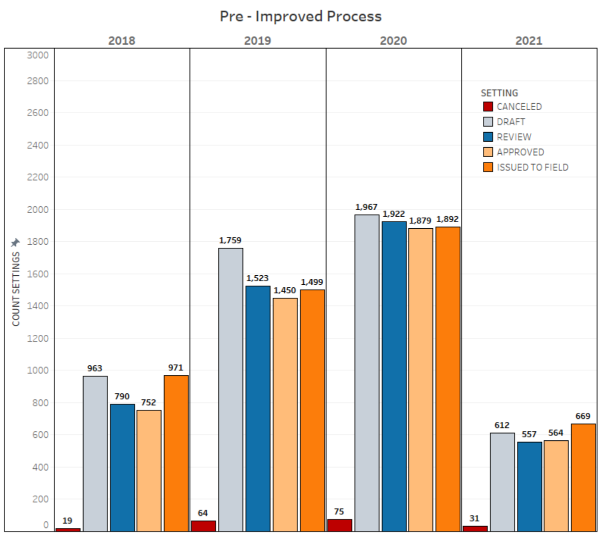

Fig. 6. Process Status Counts Prior to New Process

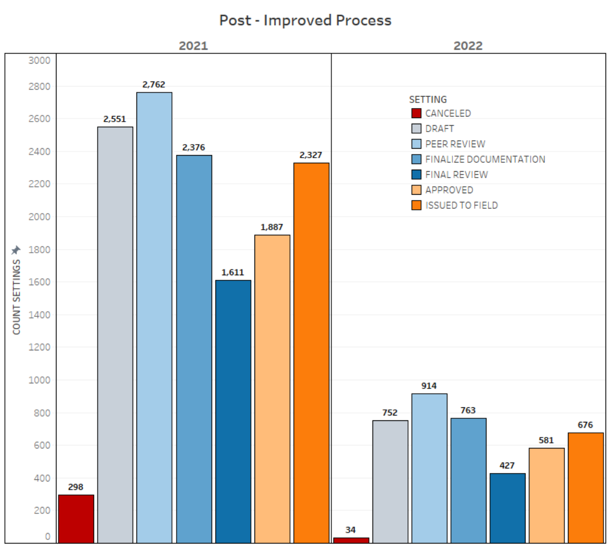

Fig. 7. Process Status With New Process

stand how long it took to complete a relay setting project from start to finish. As mentioned before, this challenge led to the *measured* requirement (Section II-A) to capture the time spent performing relay setting work. This capture of the time can be seen between the two graphs in Figure 6 and Figure 7. The setting statuses in the old process were mainly used to track the settings from the review step through the release to the field technician. This resulted in very little time difference shown between the steps as shown by the almost even counts for each step. Now that the new process covers the entirety of the project, the variation in work in a year can start to be seen in Figure 6 with the more detailed steps capturing the actual date of action.

Another challenge was to find when errors were made and their impact. One way that errors can be found is to observe relay settings that were cancelled. With the new process requiring that the workflow be maintained, it is now necessary for the engineers to cancel the relay setting row and to make a new one if modifications are made to what was previously produced. These settings were mostly cancelled due to errors that were found after the relay settings were issued and mostly due to factors outside of the relay setting process.

Comparing the number of cancelled rows between the previous process in Figure 6 and the new process in Figure 7 shows another benefit of this process; the ability to identify the times that a relay setting had to be redone. Simple errors can be identified by a change backwards in a singular step in the process and major errors can be identified by a large change in steps backwards in the process. These errors can be seen below in Figure 8 which shows the status of the relay setting in the database before it was moved to a prior status. Therefore, the *Peer Review* status having the most amount of changes to a prior status is expected, as this is where the errors should be found. The relays settings in *Final Review* before being reverted could have either been a minor error or a major error, depending on the prior status that they were moved to. More analysis into these errors will be beneficial to better understand the root cause in order to attempt error reduction.

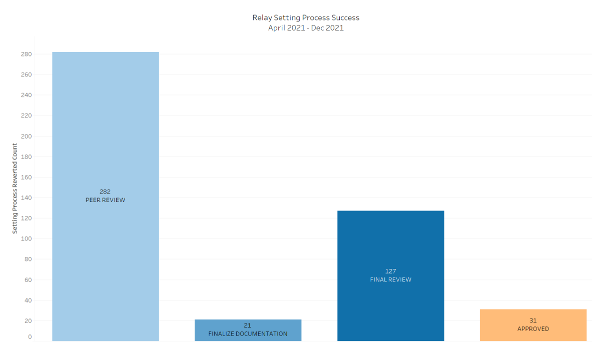

Fig. 8. Relay Settings Process Success.

B. Field Support

Another important benefit from this project is reduced time spent answering questions from the field. A necessary part of the setting engineer's job is to work with the technicians over the phone while settings are installed if anything unexpected comes up. Reducing calls due to preventable issues allows relay setters to spend more time on other tasks. Frequently, technicians will have questions regarding discrepancies between the download file and settings sheet or the test points. We anticipate the new workflow will reduce field questions, due to the more defined review process as well as automation of several steps.

There are two separate review processes that are tracked through the settings database. Each setting must go through both review processes, ensuring that each setting is looked at by at least two people, before it is issued to the field. Review job aids assist the relay setters in thoroughly checking the settings. The improved review process means that errors are caught before issuing settings, reducing required interactions with field technicians.

As discussed in Section III-D1, we automated the creation of relay download files as well as testing points. Previously, relay settings were manually typed into the download file and used the short circuit model to simulate faults and generate testing points to ensure that the protection scheme as well as settings are working as intended. Both of these provided opportunities for incorrect data to be provided to the technicians in the field, and their deployment will reduce time spent on the phone troubleshooting these types of errors.

C. Challenges

Much of this effort occurred during the early stages of the pandemic. Deployment and training of the relay settings calculator software was unexpectedly performed completely remote. Another remote challenge was faced by the settings engineer tasked with development of the Excel workbook, who was a relatively new engineer and had a basic understanding of Oncor relaying. The setter learned the complexities of Oncor philosophy in a challenging environment, as questions could not be addressed in person as usual in the remote environment.

Debugging issues in a combined software solution was also challenging, as was working with newly developed APIs for interoperablity. It was often not clear which of the three vendor packages was causing the issue or if the issue was in fact caused by an unknown security process or access policy on the deployed machines. All parties worked together not only to ensure the success of this project but also to bring greater stability to the program interfaces.

VI. Wide Area Coordination PRC-027 Requirement 2

After the integrated process was deployed for use by settings engineers, we turned interoperability efforts to wide area coordination. An initial version of the solution described here is currently being tested within Oncor. An overview of the solution's workflow is depicted in Figure 9.

The relay settings calculator drives the coordination process, after model verification has been performed in a similar manner as with the PRC-027 R1 process. Next, coordination criteria (e.g., allowable zone reach ranges, minimum CTI, etc) are specified and coordination is run on all relays in the area of study, utilizing the capabilities of the short circuit model. Violations are displayed in a filterable dashboard and can be interactively resolved in the settings calculator with verification occurring in the short circuit model. Finally, a bulk transfer of relay settings updates and associated reports are sent to affected relays' entries in the settings database.

One capability added to the setting calculator's coordination model as part of this project is the ability to compute an arbitrary symbolic mathematical expression using each relay's settings, similar to capabilities used to implement the settings philosophy template for the R1 process. The immediate use was for the wide area recalculation of loadability calculations, though we anticipate much wider user in the future.

VII. Future Work and Conclusions

Oncor has already seen significant benefits outlined in the paper both from the new settings development process as well as the software packages and their interoperability that have been combined to create an efficient, automated workfow that promotes settings accuracy. Additionally, our collaboration during thse software integration efforts yielded several avenues for exploration to further improve system protection activities:

- **Extended Data Analytics.** - We continue to evolve our data analytics around the relay setting process with the goal to better evaluate the relay setters work load. The ability to pair past project information with projected project data to forecast the amount of relay setting work will be very beneficial and allow us to more directly measure the impact of automation and interoperability efforts. With the process now showing the total time it takes to complete a settings project, the goal is to better understand the cost of full time employees completing the work compared to contract resources. This will also provide the ability to see relay setting workload over time rather than just a total count completed.
- **Automation Performance and Responsive Improvements.** One of the benefits of software interoperability is the ability to combine their interactions in a rich graphical user interface which the engineer can use to drive the process. As is common across domains, an increase in speed (e.g., short circuit calculations or report generation) enables a higher level of interaction with users, allowing them to use the software in ways previously not possible. We intend to further increase the performance of our integrated software solutions and explore the new interactions this unlocks with engineers.
- **Autonomous Verification.** Our short circuit model software will soon store the model in a repository with every change tracked akin to software source code revision control [15]. Extending this comparison, we envision the system protection equivalent of continuous integration [16], with autonomous checks that verify expected invariants, such as proper relay coordination, are not violated by new settings or other model changes. This will allow common sources of errors to be disccovered at earlier stages of the setting development process.
- **Formal Mathematical Modeling of Protection Standards.** Building upon our foundations in settings automation [1], [2] and recent work in wide area coordination auto tuning [3], [4], we continue development of formal mathematical specifications of fundamental problems in system protection. This effort is significantly aided by interoperability, as it allows us to drive simulations to create the formal models and subsequently invoke numerical solver libraries from within the same application.
- **Calculation Sheet Refinement.** - We continue to add features to the line settings spreadsheets based on user feedback. We are also developing excel based settings sheets for all of our panels and relays to take advantage of

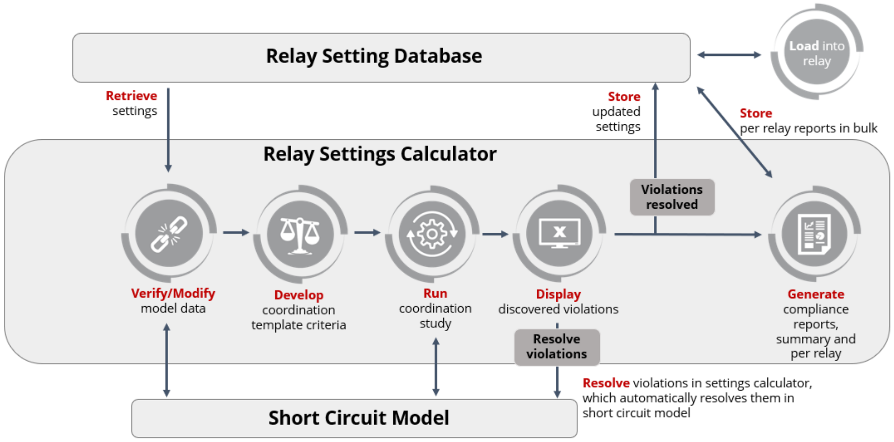

Fig. 9. Wide Area Coordination Software Process Workflow

the features in excel that process automation. The lessons learned from this project are vital in these future efforts.

We believe that the future is bright for innovation in system protection software and automation. With software interoperability, sophisticated solutions that assist throughout the entire relay settings process promise to improve the productivity of engineers and increase the reliability of the power grid by reducing the chance for errors and providing the tools to move forward despite growing complexity.

References

[1] M.J. Boecker, G.T. Corpuz, J. Perez, and L. Hankins. "Novel approach to relay setting development." 2017 70th Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2017.

[2] C. Thomas, J. Perez, L. Hankins, and H. Tribur. "An innovative and automated solution for NERC PRC-027-1 compliance." 2018 71st Annual Conference for Protective Relay Engineers (CPRE). IEEE, 2018.

[3] N. Thomas, L. Hankins and J. Perez, "Simplifying Wide Area Coordination of Directional Time Overcurrent Relays Using Automatic Settings Selection." 2019 72nd Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 2019.

[4] N. Thomas, L. Hankins and J. Perez, "Auto-tuned Solution to Wide Area Coordination Issues of Distance and Directional Time Overcurrent Relay Settings." 2021 74th Conference for Protective Relay Engineers (CPRE), College Station, TX, USA, 2021.

[5] "PRC-027-1: Coordination of Protection Systems for Performance During Faults." (2015, November). Retrieved February 22, 2018, from North American Electric Reliability Corporation.

[6] ASPEN Oneliner. https://www.aspeninc.com/web/software/oneliner.

[7] ISO/IEC. (2020). "ISO International Standard ISO/IEC 14882:2020(E), Programming Language C++." [Working draft]. Geneva, Switzerland: International Organization for Standardization (ISO). Retrieved from https://isocpp.org/std/the-standard

[8] "The boost graph library: user guide and reference manual." Addison-Wesley Longman Publishing Co., Inc., USA, 2002.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. "Introduction to Algorithms, Third Edition." The MIT Press, 2009.

[10] A. V. Aho. "Algorithms for finding patterns in strings, Handbook of theoretical computer science (vol. A): algorithms and complexity." MIT Press, Cambridge, MA, 1991.

[11] Doble Enoserv PowerBase. https://www.doble.com/product/powerbase.

[12] V. I. Levenshtein. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." Soviet Physics Doklady, Vol. 10, February 1966, p.707.

[13] Setting Automation Relay Assistant. https://synchrogrid.com/sara-software.

[14] G. Navarro. "A Guided Tour to Approximate String Matching." ACM Computing Surveys. Association for Computing Machinery, New York, NY, March 2001, pp.31–88.

[15] J.Loeliger, Jon. "Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development". O'Reilly Media, Inc., 2009.

[16] P. M. Duvall. "Continuous Integration. Improving Software Quality and Reducing Risk." Addison-Wesley, 2007.

VIII. Author Biographies

Jared Gurley graduated from Texas Tech University with a B.S. degree in Electrical Engineering in 2009. He works for Oncor as a manager in the relay setting group and is a registered professional engineer in Texas. He is responsible for managing a small team of engineers, overseeing contract relay setting resources, and working with software to improve efficiency for the relay setting team. Jared previously worked in Transmission Operations and as a relay setting engineer.

Luci Hays earned B.S. and M.S. degrees in Electrical and Computer engineering from Baylor University in 2016 and 2018, respectively. She works as a system protection engineer for Oncor in the relay standards and studies group. She helps design and develop standards, maintains existing standard setting templates and drawings, and assists in documenting philosophies and procedures. She previously worked as a relay setting engineer.

David Ridley earned his B.S. and M.S. degrees in Electrical and Computer engineering from Baylor University in 2017 and 2019, respectively. He joined Oncor's system protection group in 2019 as a relay setter. He works to create settings and troubleshoot issues with field technicians. He is part of a team to improve engineer efficiency using automation.

Zachary Austin graduated from Texas A&M University in 2013 with his B.S. in Electrical Engineering. He is a registered professional engineer in the state of Texas. He joined Oncor Electric Delivery in 2013 and is currently the PowerBase Programs Manager in the System Protection group. In this role he manages a team in the development, maintenance, and improvement of PowerBase at Oncor. Previously he spent six years as a relay setting engineer.

Nathan Thomas earned B.S. and Ph.D. degrees from Texas A&M University in 1999 and 2012, respectively, both in Computer Science. He has an extensive background in high performance computing for large-scale engineering and scientific applications. He is also interested in machine learning and how it can be used to maximize system performance. Nathan cofounded and leads development at SynchroSoft, the software and automation division of SynchroGrid.

Luke Hankins graduated from Texas A&M University with a B.S. degree in Electrical Engineering in 2015. He is a registered professional engineer in the state of Texas and works for SynchroGrid in relay setting automation. He writes software applications in various programming languages, focusing on simplifying relay setting development. Luke also performs relay settings verification and mis-operation analysis.

Alberto Borgnino graduated from the University of Genoa with a M.S. Degree in Electrical Power Engineering in 1993. He is a protection engineer with significant software development experience. He has worked creating relay setting automatic coordination systems and also has extensive experience creating relay models in short circuit simulators.

Joe Perez received his B.S. degree in Electrical Engineering from Texas A&M University in 2003. He is the author of many relay application notes and technical papers at WPRC, Texas A&M, and Georgia Tech Relay Conferences. Joe is the owner of SynchroGrid, a registered professional engineer in the state of Texas and a member of PSRC, IEEE, and PES.