PNNL-25747

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

Curated Power Grid Data Repository

Requirement & Prototype Specification Document

August 2016

P Sharma
T Gibson

MJ Rice

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

PNNL-25747

Curated Power Grid Data Repository

Requirement & Prototype Specification Document

P Sharma
T Gibson

MJ Rice

August 2016

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Contents

Figures

Tables

1.0 Introduction

Evolving power grid research and applications possess a number of underlying requirements that existing open-access datasets do not address. In order to create a data repository that supports easy data access, data sharing, model evolution, benchmarking, community involvement, data publication, and other such advancements, it is vital to define specific requirements. This document describes these requirements for a Curated Power Grid Data Repository (CPGDR) and gives detail of a bench-scale prototype that implements a subset of given requirements.

2.0 Requirements

2.1 Data Management

The CPGDR should support a variety of data models and scenarios, as well as a number of raw data types; it should also have the flexibility to accept new types as needed. First, the data repository should support the uploading, searching, and sharing of data models for transmission grids, distribution grids, and hybrid grids. Users should be provided simple mechanisms or interfaces to upload data associated with these models.

a. The repository should be able to store and manage different types of data, including user account and authorization details, power grid models, power grid scenarios, metadata associated with models and scenarios, provenance tracking, results from power grid applications, user comments, and community discussions.

b. Different types of data require storage in different type of data sources. For example, phasor measurement unit (PMU) data could be stored in a specialized time series data source. Data can be accessed through a variety of interfaces, including databases, raw files, and commercial tools. To support this, the repository should provide a flexible Application Programming Interface (API) to integrate with SQL, NoSQL, and text based data sources.

c. The repository should store power grid models in a way that can be easily transformed into another format when needed. Several types of power grid models should be supported, and each in a number of formats. This includes models for transmission systems, distribution systems, and building management systems.

d. Several types of raw data streams should also be supported (e.g., PMU data, SCADA data, and smart meter data).

e. These data sources can be associated with each other and with models. For example, a raw PMU stream can be associated with a transmission model and SCADA data from the same time period.

2.2 Data Upload and Metadata Capture

When data is uploaded to the data center, certain metadata should be captured for tracking and management purposes, including the following:

a. **User**: The user who uploaded the data.

b. **Model**: The associated model (if it is a raw data stream, the associated model should be tagged).

c. **Provenance after Editing**: The previous revision of the model and the process performed while editing.

d. **Statistics**: Statistical information on a per user or center basis, including user uploads, published models, and average model rating.

e. **Phenomena in the Data**: Data can be tagged with metadata of phenomena captured.

f. **Tags and Annotations**: The user should be able to provide tags to easily search the model based on tags. The repository should also support data annotations.

2.3 Data Quality

The repository should support an automated process to check for data quality whenever a new model or scenario is uploaded. In order to save the time and effort of every user from performing the same tasks on a dataset, the repository should be able to run quality check tools and publish the results on a web portal. These common tasks include:

a. **Data Validation by Tools after Model Upload**: In order to develop valuable tools and analytics, researchers need quality data and models. The quality of data can be enhanced through processing during upload. Some things to validate include:
   - Correct format - verify the correct format and syntax
   - Model solvability - analysis run to ensure completeness of models
   - Basic errors - check for other common model errors
   - Duplicate models - check to make sure the model isn’t already in the repository

b. **Validate and Approve Algorithms to Process Uploaded Data**: For additional validation, algorithms generated by users can be added to validate and approve data that is uploaded. These algorithms can check for erroneous readings from a sensor stream or to verify the correct format.

c. **Validation and Verification of Datasets and Dataset Creation Tools**: validate the realistic nature of automatically generated public datasets by comparing against the features and metrics of real-world datasets. Verify the dataset’s suitability for power grid optimization applications by solving optimization problems using commercial tools provided by our vendor partners.

d. **Clean up Erroneous Readings from a Senso**r. Cleaning up erroneous readings can greatly improve the accuracy of other methods, such as event detection.

2.4 Application & Tools Integration

Commonly used power grid applications and tools such as power flow solvers should be integrated with the repository and exposed to researchers via the web portal. As the power grid research is growing, the addition of new power grid applications would allow researchers to not only find these functionalities in one place but also execute these applications and publish the results via the web portal. Power grid data applications and tools are generally developed in different languages, including MATLAB, JAVA, R, Python, C#, and C++. In order to support these languages, the data repository should provide an interface to these applications. Using this interface, applications can integrate with the repository and expose their functionality via the web portal. This interface can be provided by developing a common API or by implementing web services or a combination of both.

2.5 HPC Integration

Some power grid data applications would require them to run on a high performance computing (HPC) cluster. For example, a user may upload a dataset and run a contingency analysis on it that would require an execution on an HPC cluster. The repository’s web portal should allow a user to select a model or scenario and perform a function on that data by launching the application on an HPC machine. The web portal should publish the resulting data when it becomes available from the application. As an additional feature, an email could be delivered to the user when the results are ready. Users should have the ability to view, download, and publish these results.

Many utilities do not have HPC capabilities and struggle to connect with national lab HPC capabilities due to access issues and firewall policies. Using the web portal or a hosted application in cloud computing infrastructure such as Amazon Web Services (AWS) are several options to provide HPC capabilities to utilities.

2.6 Web Portal

In order to create an easy experience uploading and interacting with models and data, a user-friendly interactive web portal or workbench should be a part of the data repository. This would help users to see what is new, find their favorites, and browse for models of interest.

a. **Capture and Display Metadata**: The web portal should also capture metadata regarding users, uploads, and curation, and allow users to view statistics based on user activity. Some statistics of interest include user uploads, downloads, publishing, and model rating/comments. In order to do so, user logins will need to be managed within the web portal.

b. **Data Uploads**: Data uploads should be supported in a number of formats. Users should be able to upload transmission and distribution models in the common formats (PTI 23/26/29/31, CIM). They should also be able to associate a PMU stream with a model, or to upload SCADA data in relation with a particular model. When uploading models, users should also be able to preview the file and use a tool to verify the model view. It would also be highly useful to tag models and data with certain characteristics, such as model type, data events, or features.

c. **Data Edits and Provenance**: Once models have been uploaded, users may want to make specific edits to the model, or alter the entire model in order to anonymize it for publication. These are edits that the web portal should support; provenance history should be tracked and made available for viewing by the appropriate users.

d. **Publish**: Another feature that a web portal should support is the option to publish models and data for public viewing after the necessary validations or edits have been made. Once data has been published, other users would be able to view the model details and or metadata.

e. **Downloads**: Users should also be able to download models or data with the desired format transformation applied. Other users should also be able to view the provenance of the model or data, starting with the time it was published.

f. **Ratings and Comments**: Users should have the ability to ask questions about models that can be answered by the creator, or the community as a whole. The concept of community participation is very important to the success of such a repository.

2.7 Data Format Transformation

Different researchers and applications work with different formats of data, including common information model (CIM), bus-branch, node-breaker, and PTI. In order for a model to be used by different applications seamlessly, it is important that the data repository supports data format transformation. For example, if a model is uploaded as bus-branch but a user requires it in a node-breaker format, then the repository should support tools to convert that model and allow the user to download the model in that desired format. The repository should be integrated with data transformation algorithms and tools that would allow a model uploaded in a specific format to be converted into a different format if required. This would also allow different power grid applications, integrated with the repository and working with different data formats, to exchange data. This would also require that data storage should be structured in a way in which data transformation can be supported.

2.8 Data Download and Multiple Format Support

The repository should provide a mechanism to download model and scenario data. This would allow users to download data to be run on an external tool. A registered user should be able to share a downloadable link of a dataset with an external user. The unregistered user should only be able to view and download the data and should not have ability to edit it in any way. The repository should allow registered users to upload results from the external tool and link it with the dataset. The result from the external tool should then become a part of the dataset provenance.

2.9 Data Publication

In order to expose the data to other researchers and scientists, users should be able to publish models, scenarios, and application execution results with a citable url. The repository should provide a digital object identifier (DOI) via a third party provider to enable data citation.

2.10 Data Curation

Data curation is required to provide researchers with models and data of known quality and to filter out inaccurate or low value data sets. Automated analysis should be run as models are uploaded to verify format correctness and consistency. Users should also have the ability to approve and add their own algorithms for data validation. In addition to this basic automated validation, the system should support a curation process by subject matter experts. Designated curators would be approved by a steering committee, and new models would be routed to the most relevant curator(s). These curators should have the ability to annotate models, define types, and organize and evolve models. They should also have the ability to archive or remove models and scenarios from active use when they are found to be inaccurate or no longer relevant. Intellectual property issues are also an area of concern, and curators should have the ability to manage these issues. This means that prior to any data dissemination; data will be reviewed per the public release guidelines established by the data repository steering committee. Venues should be identified for data dissemination; IEEE Power and Energy Society could help to identify impactful opportunities. Developers of the curated data center should also make themselves available to lecture on the topic to university power engineering classes at regional and national institutions. Social media would also be used to generate community awareness of the repository.

2.11 Provenance

One of the major requirements for the data repository is to keep track of who uploaded what data and how it changed over time and by whom. A user can edit a model or scenario and also provide new metadata associated with that dataset. The data repository should include provenance tools to keep track of such operations. Each edit should result in a new dataset with a provenance link between the original and the edited data. The web portal should provide a way to visualize such provenance by showing an interactive chart. Users should be able to move forward/backward on that chart to view exactly what is changed in a model or scenario since upload. Provenance should keep track of changes by storing metadata, such as:

- Changed By (user name),
- Changed On (date),
- Original Data, and
- Tools and Operations Used to Change Data.

2.12 User Accounts & Security

With an open-access repository, anyone should be able to download a published model or scenario, but only a registered user should be able to upload and modify a dataset. The data repository's web portal should allow a new user to register by creating an account and capturing the metadata associated with an account.

There should be proper authorization for each new user to gain upload/edit access to the repository. There should be additional authorizations needed before granting curator status. The repository should have the ability to connect to common enterprise identity services over Lightweight Directory Access Protocol (LDAP), such as Active Directory.

2.13 Performance and Scalability

Each component of the repository should be designed and implemented to achieve the necessary performance to support the number of users and models/data in the repository. The repository should have the ability to scale in order to archive an arbitrary number of power system models. High-volume web access techniques should be used for the web portal. Data storage requirements should be planned well beyond the anticipated requirements, with a planned path for expansion.

2.14 Data Archival

The repository should have policies for data archival. There should be incremental backups on a daily basis, partial backups on weekends, and full backups monthly. For daily operational purposes, the repository should keep all the data in the current and active repository location; however, eventually it should be placed in a long-term archive. This archive should be in a location that can be easily accessible if needed, but apart from the routine data location. Policies and rules for data archival should be available to the research community via the web portal.

2.15 Evolution Support

Power grid research is continually evolving. The data repository should be designed to support this evolution. The repository's interfaces and the web portal should be developed in a flexible manner to support the addition of new capabilities. These new capabilities could be an addition of a new data type, data format, data source, application, web service, security rules, curation rules, or data sharing rules.

2.16 Computing Infrastructure & Support

The data repository must have the capacity for easily expandable data storage and processing capabilities. It should allow storage capacity to be added without interfering with operations. Likewise, it should also allow for processing capability to be added without unnecessary impact. A support team should be in place to monitor systems and be available to make repairs when necessary. A certain percentage of 'up-time' should be guaranteed. The support staff should also be responsible for regular system maintenance, such as backups and patching. Any critical vulnerability should be patched in a timely manner and all system resources should be kept up to date with the most recent approved version. A reliable backup system, as discussed previously, should be in place for all managed data and applications to prevent data loss in the event of a system failure.

For power grid model and data curation systems should have a minimum of 10 TB HD space and processing capability available. The data storage should support petabytes of data and the capability to balance processing loads across multiple computing resources. This storage must be sufficient to store both data models and several types of data streams, including PMU and SCADA streams. For example, 20 channels of PMU data require 5.5 GBs of space per day. Because the curated data center must be accessed externally, an external (to the organization) facing web server should be available that connects to the data center.

2.17 Synthetic Scenario Creation

The ability to generate synthetic, but realistic scenario data for use in research applications is important. This data allows researchers to develop cutting edge algorithms and applications, without having to worry about the sensitivity of real data. The data repository should support tools to generate synthetic data for scenarios. Users should be to select a model on the web portal and provide additional information to generate a scenario based on that model. When generating synthetic data, the system needs to support the entry of a number of parameters, such as weather or time, that will affect the data that is generated. Users should also be able to add embedded events to the synthetic data. The application that generates the data also needs to know what types of events the user would like to see in the generated data and when they should occur. More parameters should be researched and added to the synthetic data creation for better precision in data generation that can be used with a power grid application.

2.18 Converting Aggregated Data in to Raw

There are times when the data a researcher is able to get for describing a power grid operation is an aggregated data set. While this allows for compressed information sharing, it does not provide sufficient inputs into analysis software. Thus, tools are needed to expand aggregated data back into raw data for testing and analysis. This must be done with great caution, as what is being created is a representative data set and not an actual data set, and therefore will not contain any outliers that aggregation statistics do not describe.

2.19 Ensemble of Power Flow Cases

Sometimes researchers need to run different power flow cases with different scenarios to come up with a scenario that solves a particular problem. The data repository framework should allow researchers to run such differing cases seamlessly. Using the web portal, a user should be able to select different models and scenarios and, with some, run conditions. The data repository framework should run these cases and notify the user when the run cases are completed and can be compared. This would allow the user to run thousands of cases simultaneously to solve a problem. For example, one might want several different wind generation output settings in power flow to see how the system is behaving or determine the probability of violations occurring in the system. The framework must allow the researcher the ability to tweak power flow inputs in case of error or unexpected statistical results.

2.20 Documentation

Up-to-date documentation should be provided for both users (general and curator) and developers. It should contain 'how-to' steps for using various features of the repository. It should also explain customization needed in tools, applications, and algorithms in order to ingrate them in the framework. User documentation would allow users to follow the steps and create an account, access data, publish, and share models. Developer documentation would help application developers to integrate their code into the repository framework for further use.

3.0 Prototype Implementation

This section explains a bench-scale prototype implementation of a subset of the requirements described in section 2.0. This bench-scale prototype was developed by leveraging the open source GridOPTICS Software System (GOSS) developed at PNNL under the Future Power Grid Initiative (FPGI).

3.1 Introduction

GOSS is a software framework for integrating a collection of software tools developed by PNNL's FPGI as a coherent, powerful operations and planning tool for the power grid of the future. GOSS enables plug-and-play of various analysis, modeling, and visualization software tools to improve the efficiency and reliability of the power grid. To bridge the data access for different control purposes, GOSS provides a scalable, lightweight event processing layer that hides the complexity of data collection, storage, delivery, and management. The design of GOSS allows developers to quickly and easily stand up new integrated applications. This allowed for the development of a bench-scale prototype that meets many requirements identified. Figure 1 shows the overall architecture of GOSS.

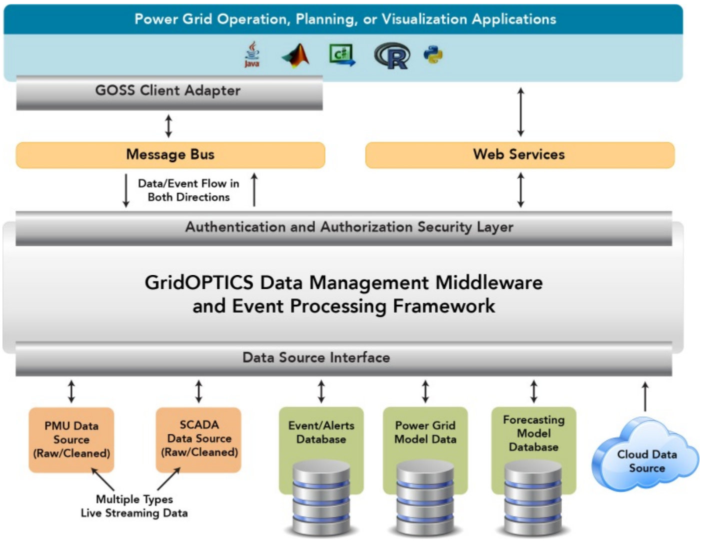
Power Grid Operation, Planning, or Visualization Applications
Java
GOSS Client Adapter
Message Bus
Web Services
Data/Event Flow in Both Directions
Authentication and Authorization Security Layer
GridOPTICS Data Management Middleware and Event Processing Framework
Data Source Interface
PMU Data Source (Raw/Cleaned)
SCADA Data Source (Raw/Cleaned)
Event/Alerts Database
Power Grid Model Data
Forecasting Model Database
Cloud Data Source
Multiple Types Live Streaming Data

Figure 1. GOSS Architecture

GOSS architecture consist of many capabilities required by the data repository framework, such as integration with SQL, NoSQL databases that contain metadata and data for power grid models, scenarios, and users; power grid model transformation between some formats; API for data access and storage; user account creation and authorization; API for integrating applications and tools; and web services support for the web portal.

3.2 Which Requirements are Met?

This section is a tabular view of requirements with remarks of their support in GOSS capabilities.

Table 1. GOSS Capability Evaluation

Requirement	Capability
Data Management	Supports the upload and management of several formats of power grid models and the storage/delivery of PMU data
Data Upload and Metadata Capture	Model uploads capture user, upload time, and model type
Data Quality	Validation performed to ensure correct format of power grid models during upload
Application & Tools Integration	External applications can be integrated using APIs in a number of computing languages (R, MATLAB, C#, C++, Python, Java, etc.
HPC Integration	Other applications have been integrated through GOSS to run on an HPC cluster
Web Portal	A simple portal developed for uploading and managing power grid models.
Data Format Transformation	Power grid models can be uploaded in one format, and downloaded a another format
Data Download and Multiple Format Support	Supports conversion between some power grid model formats; GOSS's web services can be extended to include download feature in multiple formats
Data Publication	GOSS's web services can be extended to support publication feature
Data Curation	Not developed
Provenance	Not developed
User Accounts & Security	User account and security is supported; has to be extended to be used via web portal

Requirement	Capability
Performance and Scalability	Though Apache ActiveMQ’s performance results are easily available online, performance and scalability testing for GOSS added capabilities must be done
Data Archival	Not developed
Evolution Support	Not developed
Computing Infrastructure & Support	Not applicable
Synthetic Scenario Creation	Not developed
Converting Aggregated Data to Raw	Not developed
Ensemble of Power Flow Cases	Not developed

4.0 Conclusion

We hope that this document is useful as a catalyst in discussions regarding the need for a curated power grid data repository. As more and more powergrid data are being generated and tools are being developed to help manage our grid resources and assets, the need for a curated power grid data repository will grow. This document has explored the fundamental functional requirements for creating a curated power grid data repository and compared a PNNL-developed tool (GOSS) against these requirements. GOSS is not yet a comprehensive tool, it supports a number of the requirements defined in this document, but additional effort is required to fully meet the requirements.

Pacific Northwest
NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY

Proudly Operated by **Battelle** *Since 1965*

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov