An Electric Power System Energy Management Platform (EMP) Research Testbed

Ikponmwosa Idehen, Thomas Overbye
Electrical and Computer Engineering
Texas A&M University, College Station, TX, USA
{iidehen, overbye}@tamu.edu

Loren Klemesrud
Power Systems SCADA Engineer
loren3737@gmail.com

***Abstract*—This paper presents an overview of the development of an Energy Management Platform (EMP) research testbed for power system monitoring and control. In this work, a testbed utilizes a simulator to mimic power system operations on a synthetic network. By emulating field devices, the simulator is used to further transmit grid data over a supervisory control and data acquisition (SCADA) system to an EMS software. Finally, through the design of oneline diagrams, a node-branch layout algorithm is proposed to develop interface substation displays for controlling the power system networks being used in the testbed.**

Index Terms— energy management system, SCADA, substation layout, oneline, testbed, distributed network protocol

I. Introduction

The rapidly-evolving electric grid is increasing the need for integrated platforms which aid strategic grid monitoring and control. Increasing proliferation of renewables, changing market dynamics, intricate grid control dynamics, and increased customer participation in demand management programs in large interconnected systems lead to complex power system operations. The deployment of modern sensing and control technologies, in addition to analog devices, however enables wide-area health monitoring and operator interaction with the grid [1-6].

An effective translation of research methods onto actual field operations will require simulation testbeds which validate these techniques using similar grid complexities and end-to-end monitoring and control tools. Here, dynamic simulations for different scenarios of a power system in operation are used to provide grid measurements in addition to individual component status and topology information to a grid-monitoring system. Through information gathering and processing, user-initiated test commands can then be sent to control the grid. The simulation of synthetic grids, infused with dynamic component models [7-9] on the testbed affords the opportunity to obtain realistic, real-time power system measurements. Similar to monitoring platforms in power utility firms, an Energy Management Platform (EMP) – comprising of a Supervisory Control and Data Acquisition (SCADA) system connected to Energy Management System (EMS) - monitors the state of the operating grid [10]. Coupled with other sub-systems implementing programs such as state estimation, automatic generation dispatch, contingency analysis, voltage stability and dynamics, operating grid variables can then be further controlled by issuing command signals to optimally control the operation of the grid.

This paper presents the architecture and configuration of an EMP system being designed in the research control room of Texas A&M University. The purpose of this paper is to develop an underlying framework using commercially available software tools for providing experience in power grid monitoring and operations. In turn, this would provide a platform for researchers to test innovative applications for implementing real-time monitoring, performing analytical system studies, and controlling the state of power systems.

The rest of the paper is organized as follows. Section II briefly describes a typical SCADA infrastructure layout. The implementation of the EMP testbed is discussed in Section III. Section IV presents a placement algorithm that has been used to layout substation buses (or nodes) and branches in a oneline display diagram. A cut section of a sample substation bus-branch layout is shown in Section V, followed by concluding remarks and future works in Section VI.

II. Power System SCADA Infrastructure

SCADA telemetry and data acquisition infrastructure in the operation of power systems enables the collection of information from remote substations to a central control site [10-12]. Aggregated analog measurements and binary device status information from power grid sensors are transmitted through intelligent electronic devices (IEDs) such as remote terminal units (RTUs) to a control site. Using EMS human machine interface (HMI) technologies which provide information display and analysis, command signals from a master control unit (MCU) are further sent to field devices which alter the state of the system.

A bi-directional data exchange between different devices and sites for grid monitoring and control is achieved with the aid of established communication platforms and protocols which transmit these data [10]. Several communication protocols have been known to exist for implementing power system SCADA data transmission, however the work in the testbed has been restricted to the use of distributed network protocol version 3 (DNP3) [13-16].

DNP3 is a reliable and robust protocol that is widely-used power system industry for SCADA data transmission and control [14]. It possesses features such as event- and priority-reporting which ensure master devices to be promptly updated when significant system events occur in the system, and assigns different report rates to measurement classes to

978-1-7281-4436-8/20/$31.00 ©2020 IEEE

enable more efficient usage of communication bandwidth. The protocol forwards sensor data from an outstation (e.g. RTUs) to a master device (e.g. central SCADA controller). A key feature is its ability to model an analog, binary or pulse measurement as a uniquely identifiable physical or logical entity. These points are modeled in outstations, and belong to either of five different point types depending on their attributes or functionality. These point types can either be input binary, input analog, input counter, output binary or output counter.

III. Testbed Implementation

A Dispatcher Training Simulator (DTS) provides a simulation environment for operator training, new product feature testings and demonstrations. In this work, the DTS software is deployed to serve as the EMP environment. A replay or simulation component provides power flow states from the operation of a test power system consisting of a network model, component electrical models and dynamics. SCADA measurements from this system are then transmitted to an EMS component where several sub-systems, such as SCADA, system alarm–handling, generation dispatch and contingency study applications can be used to provide information for efficient grid control.

A. Synthetic Power Systems

In the testbed, an interactive power systems simulator - PowerWorld Dynamic Studio (DS) [17, 18] - performing stability studies on synthetic grids, and implemented for a transient stability timeframe, has been used to replace the simulation component of the DTS. These synthetic grids come with in-built dynamic models for different electric components [9]. During simulation, realistic and real-time SCADA data are transmitted to the EMP thus eliminating the need for a network model. While DTS protocol communication has been successfully tested with a 200-bus system, the goal is to implement a synthetic 2,000 (or 2K)-bus Texas power system on the testbed.

The synthetic Texas system is made up of several electrical components which includes 544 generators, 1,350 loads, 157 switched shunts and high voltage transmission branches distributed across 1,250 substations.

B. Proposed SCADA Telemetry Model

Fig. 1 shows a SCADA telemetry model for simulation in the testbed.

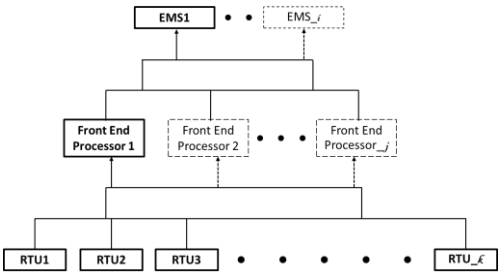

Fig. 1. High-level flowchart summarizing content creation in source file

The telemetry model replicates actual comunication that would exist between actual substations and a utility control center. Real-life EMS platforms are classified as high availability systems, hence the redundant components at the procesor and EMS levels. Front-end processors (FEPs) read and validate measurements from geographically-dispersed RTU locations, which are then mapped to existing electrical components already modeled in a SCADA database located in the EMS.

C. Architecture

The EMP-SCADA communication achitecture that is currently being set up in the testbed is shown in Fig. 2.

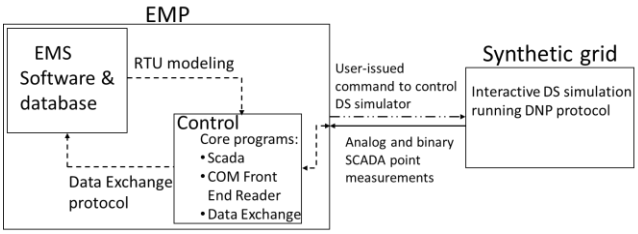

Fig. 2. High-level flowchart summarizing content creation in source file

In the synthetic grid block where the operation of the power system is simulated, the capability of PowerWorld Simulator [19] is leveraged to emulate multiple RTUs reporting SCADA-type measurements to an EMP. These measurements, either in the form of analog or binary state data, are obtained from RTUs or outstations, and transmitted as DNP3 input points. The EMP block simulates the operations of the FEP/EMS as shown in Fig. 1. A set of core programs in a control FEP serve as SCADA server application from which telemetry commands are issued to and from outstations in the test power system. A data exchange protocol performs an intra-control center communication by replicating data from the FEP to the EMS sub blocks.

D. Modeling in the SCADA Subsystem

A SCADA subsystem contains a description of the real-time data acquisition and control capabilities of the system. Upon validation at the front-end processor, SCADA measurements are transmitted and stored in the core database of the subsystem. The data structures for the communications system among host sites, data retrieval system through which SCADA measurements are acquired from RTUs, and the power system containing descriptions of the substation and different power system components are also contained in this database.

Fig. 3 shows logical hierarchy data structures (also known as branches) that have been used for host communication, data retrieval and component data modeling in the subsystem.

FEP	RTU	STATION
- CHANEL	- SCANGRP	- EQGRP
- - PORT	- - CARD	- - DEVICE
- - - MULTIDR	- - - DI/AI	- - - MEAS
(a)	(b)	(c)

Fig. 3. Data structures for DTS communication and component data modeling

The FEP data structure in Fig. 3a defines a SCADA site and provides the means for communicating with RTUs. Here, communication parameters such as data transfer baud rate, communication front end (CFE) type and communication IP and port address are configured. With the aid of a multi drop (MULTIDR) field, the FEP is set to connect with a dedicated RTU, or multi dropped RTUs [10]. The logical RTU structure is in Fig. 3b. Here, parameter definitions such as

RTU ID and address, SCADA communication protocol, scan address, rate and type, communications card address, point measurement index, deadbands and value high/low limits are set. Finally, an analog/digital (DI/AI) record field ties the RTU to the assigned substation data structure, STATION in Fig. 3c. Here, information pertaining to the substation name, area of responsibility and control, power system equipment grouping (EQGRP) and device, type of measurement record (e.g. point, analog, or control) are provided.

A benefit of using the hierarchical data structure in Fig. 3 is that measurement points transmitted to the SCADA subsystem can be uniquely identified, while mapping them to the appropriate component models. Table 1 shows some of the configuration parameter values that have been used.

Table 1. Configuration parameter values

Parameter	Value
FEP	FEP_TAMU
CHANEL	CHN_TAMU
CfeType	WINSTRMS
PORT	PTH_TAMU
Baud	9,600
Primary Address	10.110.21.31
Port Number	20,000
RTU	E.g. **Sub1**

E. Modeling DNP3 Points in the Power System Simulator

In a recent version of the software, DNP3 options have being configured in PW simulator, and a protocol. Here, the interactive simulation software for the power system is used to emulate a set of field outstations (or RTUs as modeled in the SCADA subsystem) containing point measurements.

A configuration of DNP3 points and outstations for any case is performed by navigating to the 'Tools and Add Ons' section of the model explorer after which outstations can be inserted in the DNP3 tool option. For ease of coordination, we have modeled each substation as its own DNP outstation, and further mapped it to unique RTUs modeled in the DTS. Thus, a total of 1,250 outstations have been emulated to report SCADA measurements from the test system.

An AUX file containing 23,467 number of points measured across 1,250 outstations was uploaded into the simulator for a fast and efficient DNP3 configuration. Table. 2 shows a sample DNP point list generated for a outstation (or PW substation).

Table 2. Sample point list configurations at an outstation

Outstation #	Point Object ID	Variable	Point Type	Event Class	Point ID
1143	Bus '7422'	VPU	Analog i/p	0	0
1143	Bus '7422'	VANGLE	Analog i/p	0	1
1143	Bus '7422'	FREQHZ	Analog i/p	1	2
1143	Gen '7422' '1'	MW	Analog i/p	0	3
1143	Gen '7422' '1'	STATUS	Binary i/p	2	0

The power system simulation solver generates lots of input measurements, configured as DNP points, to send to the EMP. Analog points include bus voltage magnitude, angle and frequency; real and reactive powers flowing across branches; real and reactive powers of generators and loads, and shunt reactive power. Binary status information of all electrical components (loads, generators, branches and shunts) are also streamed to the EMS. Currently, only input (i/p) point types have been set, and points assigned to any of three event classes. The values of points assigned to class 0 are only updated when periodic integrity scans are run on the FEP server side. Their current values are not stored at their outstations, and also not reported during an event. In contrast, values of points in classes 1,2 and 3 are constantly updated and reported to the server when events occur. Their outstations ensure that real-time changes are consistently communicated to the server. Finally, point IDs in the sixth column uniquely identifies points in their point type arrays.

IV. Substation Node-Branch Visual Display

Control center HMI technologies presenting grid data require substation oneline displays to help users visualize and supervise the system [20-24]. A manual rendition of individual substation oneline diagrams in the DTS is neatly and easily done for a small system, however it becomes infeasible in large systems that consist of several substations and attached components. Also, considering various complexities in laying out components in oneline diagrams, the time and mental effort involved in rendering unique substation displays may prove to be a challenging task. In this paper, we develop an algorithm to automatically layout substation buses (or nodes), breakers and their branch connections. It assigns specific orientations to substation components, avoids duplicated renditions and branch crossings while neatly positioning all substation components. The method ensures a common outlook of all substations in the DTS, and minimizes any post-rendition task that may be required to adjust unique substations.

A. Display Source Configuration File

The EMP consists of a display setup program where substation displays can be created. Here, several display components are provided to users to allow for customization of user interfaces. With the aid of linkages connected to application databases, users are then able to access and interact with application data, such as SCADA data, through the designed interfaces. A display source file contains all the configuration data of primitives and element sets written in ASCII text strings.

In this work, the method to create node-branch oneline displays for the synthetic system initially executes a bus-branch layout algorithm for each substation (STN), and after which the corresponding ASCII strings are written to a source configuration file. Fig. 4 is a summary of the content creation process for the source file.

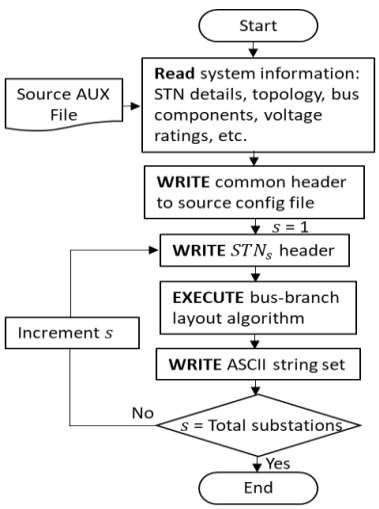

Fig. 4. High-level flowchart summarizing content creation in source file

An auxiliary file (*.AUX) used by PowerWorld which is known to store simulation data for a synthetic network serves as a source document from which all case and model data are extracted. It contains all information of the power system case, such as connected system loads, generators and shunts (hereafter known as bus components), substation details, branch toplogy and nominal voltage ratings. The use of ASCII text strings of groups of pictures or elements in a common file header avoids the repetition of common objects a user may intend to render across all substation displays.

B. Substation Node-Layout Algorithm

An overtly perfect rendition of all components may not be feasible on a substation oneline diagram. However, the proposed approach uses a method to simplify the layout of bus-breaker-branch connections, while minimizing the amount of post-rendition efforts that will be needed to make manual adjustments in each substation diagram.

Table 3 is a list of the common power system components and symbols that have been used in the display setup program. The color used to render color-coded components is based on the nominal voltage of the bus or branch to which the component is connected.

Table 3. Power system component symbols

Component	Color-coded?	Symbol
Bus	Yes	
AC line	Yes	
Transformer	yes	
Load (Feeder)	Yes	Feeder
Shunt	No	
Generator	Yes	G
Breaker	No	
Switch	No	
Formatted field placeholders	No	XXXXXXXX I Q 9999 KV
External substation	No	BBBBBBBB

In any substation, a prior step is the identification of the most-connected bus, and defined as the bus with the highest branch connections with other buses belonging to the same substation. This is designated as the substation main bus, R_{bus} from which propagating branches connect to non-neighbor or neighbor nodes and the rest of the substation components. In this context, a neighbor is a directly-connected bus in the same substation, while a non-neighbor belongs to another substation.

C. Depth-Search Technique

This method involves an apriori consideration of the depth level or extent of the descendants of a neighbor node before drawing a branch connection to the node. A depth-search technique used in graphical tree structures [25-27], the procedure determines the amount of horizontal spacing required to position a next adjacent neighbor node to the bus.

Fig. 5 shows a flow-chart describing the process for assigning coordinate values in a two dimensional system, and rendering components in the oneline diagram for a substation (*STN*).

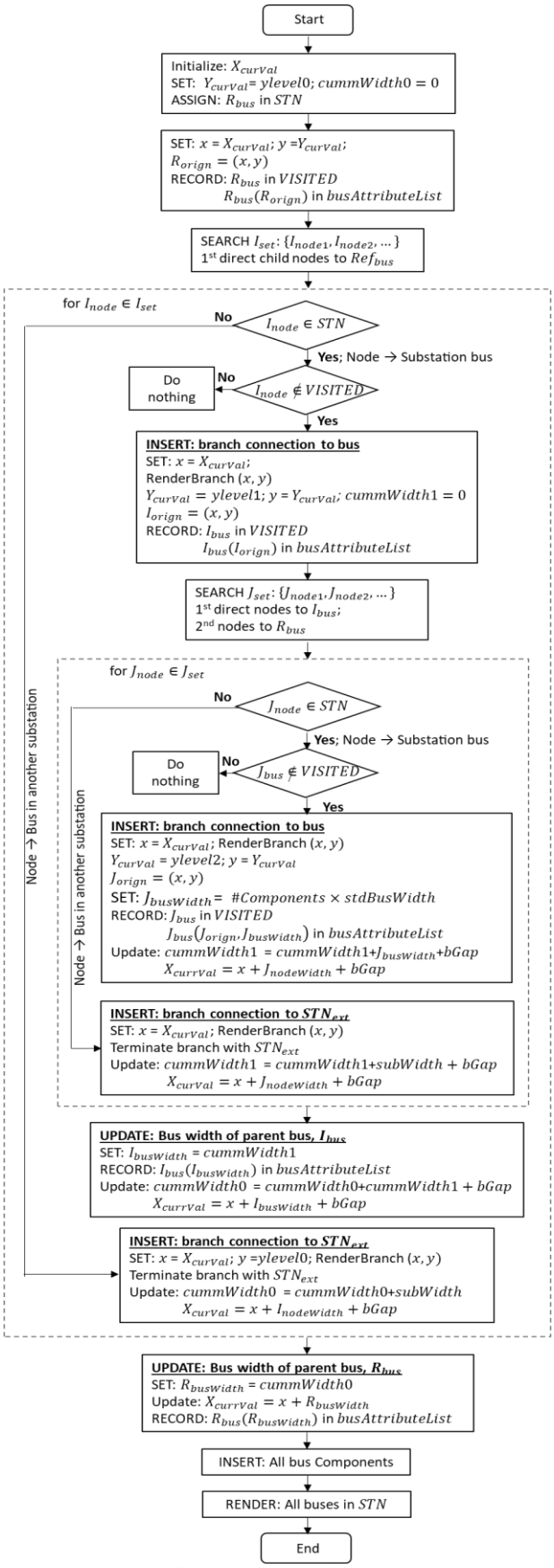

Fig. 5. Flow chart describing coordinate assignment and rendering of substation components

In the figure, a prior assignment is the determination of levels (or layers) of vertical coordinates, $ylevel$ to be used. For illustrative purpose, three levels have been assumed i.e. $ylevel$ 0, 1 and 2 where R_{bus} is in the topmost level, $ylevel0$.

At a reference bus, the left terminal position of its bus component symbol is set as the value of a current X-position variable, X_{curVal}, while the $ylevel$ of the bus is set in a current Y_{curVal}, variable. The depth-search is performed by using the system topology data to identify all unvisited, directly-connected child nodes to be positioned at an immediate lower $ylevel$ to the reference bus. If a non-neighbor is identified, it is immediately rendered by drawing a descedant branch component object (i.e. transformer or AC line depending on the nominal voltages of both nodes) to an external substation component, STN_{ext} The leaf node of that branch is considered to be reached, and X_{curVal} is updated by incrementing it by a substation interval, $subWidth$ and a bus separation, $bGap$ before the next adjacent child node connection is considered.

In the event that a neighbor is detected, X_{curVal} and the lower $ylevel$ are assigned to that node and stored in a $busAttributeList$ table object while the node is marked as a visited node. At the X_{curVal} position, a descendant branch component is then drawn from the reference bus $ylevel$ to the child node lower $ylevel$. However, no terminating bus is attached. The value of Y_{curVal} is updated to the lower $ylevel$ where the child node is located, and the depth-search process is re-initiated. In a downward search, this process continues until a leaf node (i.e. bus whose neighors have all been visited or a non-neighbor) is reached thus generating a series of tree nodes. This is followed by an upward search, or backtracking, where Y_{curVal} is set to the Y-position of a successive parent node, and the value of X_{curVal} updated by incrementing by the sum of its bus width and $bGap$ before the next adjacent child node connection is considered. .

The above-described process for drawing the substation oneline diagram possesses similar with the recursive process used in pre-order traversal algorithms when drawing graphical tree structures [27, 28]. However, a requirement for subtree symmetry has been ignored since by definition, nodes would imply buses with variable bus widths depending on the number of connected bus components. Also, the process of centrally rendering multiple branch connections between buses, while avoiding cluttering of component symbols proved to be an arduous task.

In the Fig. 5, the method for setting the width of a bus is based on its position in the graphical tree. If the bus is observed to be located in a leaf node, a provisional buswidth is determined by the number of connected components at the bus (branches inclusive) multiplied by a pre-defined parameter, $stdBusWidth$. Otherwise, the bus is a parent node whose width is obtained by aggregating the bus width of its different child nodes (neighbor buses and non-neighbor external substations inclusive).

D. Rendering Bus-Connected Electrical Components

Prior to rendering all substation buses in the flow chart in Fig. 5, all electrical components such as loads, generators and shunts need to be inserted at their respective buses. This process is carried out by using bus information stored in the AUX file, and leveraging bus node details already saved in the $busAttributeList$ object.

In Fig. 6, a summation of the x-origin and width of the bus determines the position where a first bus component should be rendered after which these most recent values are stored in a $busCurrentXYList$ object.

First bus component
Obtain busOrigin for the bus from $busAttributeList$
$x_{component}$ = busOrigin.x + busWidth
$y_{component}$ = busOrigin.y
insert_Bus_Comp($x_{component}$, $y_{component}$)
Record: ($x_{component}$, $y_{component}$) in $busCurrentXYList$

Additional bus components
Obtain busCurrrentPos for the bus from $busCurrentXList$
$x_{component}$ = busCurrentPos.x
$y_{component}$ = busCurrentPos.y

$x_{component} = x_{component} + cmpGap$
insert_Bus_Comp($x_{component}$, $y_{component}$)

Update: busCurrentPos.x = $x_{component}$
Record: busPosition in $busCurrentXList$

Fig. 6. Rendering bus components

Subsequent retrieval of the available current bus positions from $busCurrentXYList$ while factoring a component gap spacing $cmpGap$ further ensures the insertion of additional components at the bus.

E. Substation Bus Rendering

A final step in generating the oneline diagram using the flow chart in Fig. 5 is to render all the buses in the substation after all connected bus components have been drawn. Here, a horizontal orientation applies to all bus sysmbols, and it is important that bus overlapping is minimized. A direct method to achieve this is to connect both left and right terminals of the bus stored in the $busAttributeList$ and $busCurrentXYList$ objects respectively.

Obtain busOrigin for the bus from $busAttributeList$
x_1 = busOrigin.x
y = busOrigin.y

Obtain busCurrrentPos from $busCurrentXList$
x_2 = busCurrentPos.x

insert_Bus(x_1, x_2, y)

Fig. 7. Rendering a bus

x_1 is the originating bus position on the left, x_2 is the most recently occupied position by an electrical bus component, and y is the vertical layer where the bus is located. This method of directly connecting x_1 and x_2 avoids overshooting of buses of either leaf or parent nodes. In a case when no component is attached at a bus (besides outgoing branches), both x-values coincide, and the resulting bus degenerates to a point. By setting a non-zero value as an alternate default width, this instance of zero bus width is easily prevented.

Following the bus rendering process, optional descriptive texts, such as bus names or IDs, can be inserted to improve user's recognition 'of the substation.

V. Sample Substation Display

A section of a oneline diagram of one of the substations in the 2k-bus system, and generated using the steps in the flow chart is shown in Fig. 8. Besides outgoing non-neighbor connections at the main bus (ID:8082), it shows other branches descending to different nodes associated with the substation. Here, a graph structure with five vertical layers adequately captured all the components and branches. The colors at separate nodes are used to differentiate among nominal voltages existing at different buses whose varying width can also be observed. Bus width for ID:8086 is based on the single load (feeder) object component attached at the bus, while the variable-widths of the other buses are based by their child buses.

Table 4 shows bus IDs, order of visiting and rendering nodes, and the order of recording width values of neighbor buses.

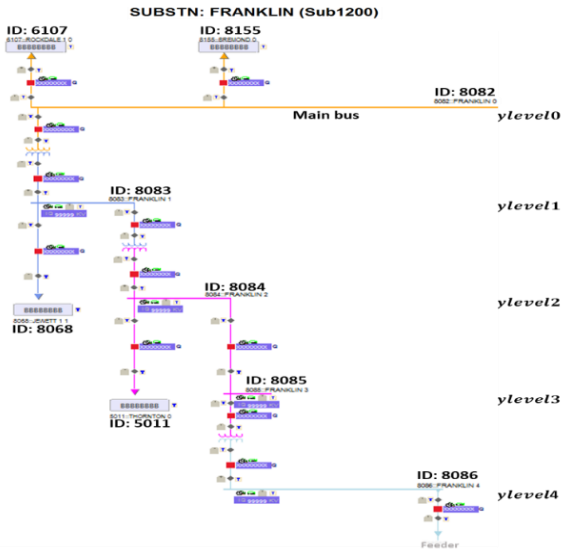

Fig. 8. Cut section of oneline diagram for substation (Franklin)

Table 4. Order of visitation, bus width recording and rendering

Bus ID	In-substn?	Volt (kV)	Visited	Rec: Width	Render
8082	Yes	500	1	5	5
6107	No	500	2	Nil	1
8155	No	500	3	Nil	2
8083	Yes	230	4	4	5
8068	No	230	5	Nil	3
8084	Yes	161	6	3	5
5011	No	161	7	Nil	4
8085	Yes	161	8	2	5
8086	Yes	115	9	1	5

The depth-search algorithm determines the visitation order at each node, and further establishes the instance when an x -coordinate is assigned to a node starting with the main bus (ID: 8082) at visitation order 1. At this time, only branch connections and non-neighbor substation objects are drawn. Hence, the rendering order for external substations are nodes with IDs 6107-8155-8068-5011. In a reverse order, the bus width of neighbor buses are recorded starting from the leaf bus (ID: 8086) at the lowest *ylevel*. Finally, in a separate process (with rendering order 5), all buses are then drawn.

VI. Conclusion and Future Work

In this paper, we have presented the development of an energy management research testbed for testing future innovative grid monitoring applications. A synthetic power grid emulating the presence of several field RTU devices is used to deliver SCADA measurements to a dispatcher training simulator (DTS) implementing an energy management platform for grid monitoring. Furthermore, we have proposed a simple and easy to implement node-branch algorithm to position bus and branch components in the design of substation oneline graphical interfaces.

A long-term goal is to provide a platform for user-experience implementing real-time power system operation and control. In an ongoing work, DNP3 output points for enabling the user-issued commands for grid control is being tested. Future work will also focus on expanding the proposed telemetry model in a fully licensed EMS software for educational use, and integrating critical subsystems, such as state estimation and generation dispatch to the testbed.

References

[1] A. Bose, "Smart Transmission Grid Applications and Their Supporting Infrastructure," *IEEE Transactions on Smart Grid,* vol. 1, no. 1, pp. 11-19, 2010.

[2] J. D. Taft, "Grid Architecture: A Core Discipline for Grid Modernization," *IEEE Power and Energy Magazine,* vol. 17, no. 5, pp. 18-28, 2019.

[3] M. Asano, "Hawai'i's Grid Architecture for High Renewables: Developing the State's Modernization Strategy," *IEEE Power and Energy Magazine,* vol. 17, no. 5, pp. 40-46, 2019.

[4] O. Ellabban, H. Abu-Rub, F. J. R. Blaabjerg, and S. E. Reviews, "Renewable energy resources: Current status, future prospects and their enabling technology," vol. 39, pp. 748-764, 2014.

[5] Z. A. Vale, H. Morais, M. Silva, and C. Ramos, "Towards a future SCADA," in *2009 IEEE Power & Energy Society General Meeting*, 2009, pp. 1-7.

[6] F. F. Wu, K. Moslehi, and A. Bose, "Power System Control Centers: Past, Present, and Future," *Proceedings of the IEEE,* vol. 93, no. 11, pp. 1890-1908, 2005.

[7] T. Xu, A. B. Birchfield, and T. J. Overbye, "Modeling, Tuning, and Validating System Dynamics in Synthetic Electric Grids," *IEEE Transactions on Power Systems,* vol. 33, no. 6, pp. 6501-6509, 2018.

[8] T. Xu, A. B. Birchfield, K. S. Shetye, and T. J. Overbye, "Creation of synthetic electric grid models for transient stability studies," in *The 10th Bulk Power Systems Dynamics and Control Symposium (IREP 2017)*, 2017.

[9] Electric Grid Test Case Repository [Online]. Available: https://electricgrids.engr.tamu.edu/electric-grid-test-cases/

[10] M. S. Thomas and J. D. McDonald, *Power system SCADA and smart grids*. CRC press, 2017.

[11] S. A. Boyer, *SCADA: supervisory control and data acquisition*. International Society of Automation, 2009.

[12] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye, and D. Nicol, "SCADA cyber security testbed development," in *2006 38th North American Power Symposium*, 2006, pp. 483-488: IEEE.

[13] K. J. D. U. G. Curtis, "A DNP3 protocol primer," vol. 2005, 2005.

[14] "IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)," *IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010),* pp. 1-821, 2012.

[15] T. Mander, R. Cheung, F. J. c. Nabhani, and security, "Power system DNP3 data object security using data sets," vol. 29, no. 4, pp. 487-500, 2010.

[16] S. Mohagheghi, J. Stoupis, and Z. Wang, "Communication protocols and networks for power systems-current status and future trends," in *2009 IEEE/PES Power Systems Conference and Exposition*, 2009, pp. 1-9.

[17] T. J. Overbye, Z. Mao, K. S. Shetye, and J. D. Weber, "An interactive, extensible environment for power system simulation on the PMU time frame with a cyber security application," in *2017 IEEE Texas Power and Energy Conference (TPEC)*, 2017, pp. 1-6.

[18] T. J. Overbye, Z. Mao, A. Birchfield, J. D. Weber, and M. Davis, "An Interactive, Stand-Alone and Multi-User Power System Simulator for the PMU Time Frame," in *2019 IEEE Texas Power and Energy Conference (TPEC)*, 2019, pp. 1-6.

[19] PowerWorld. PowerWorld Simulator [Online]. Available: https://www.powerworld.com/products/simulator/overview

[20] A. B. Birchfield and T. J. Overbye, "Techniques for Drawing Geographic One-Line Diagrams: Substation Spacing and Line Routing," *IEEE Transactions on Power Systems,* vol. 33, no. 6, pp. 7269-7276, 2018.

[21] T. J. Overbye, E. M. Rantanen, and S. Judd, "Electric power control center visualization using Geographic Data Views," in *2007 iREP Symposium - Bulk Power System Dynamics and Control - VII. Revitalizing Operational Reliability*, 2007, pp. 1-8.

[22] T. J. Overbye and J. D. Weber, "Visualization of power system data," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 7 pp.: IEEE.

[23] P. Cuffe and A. Keane, "Visualizing the Electrical Structure of Power Systems," *IEEE Systems Journal,* vol. 11, no. 3, pp. 1810-1821, 2017.

[24] Y. Zhu and O. P. Malik, "Intelligent automatic generation of graphical one-line substation arrangement diagrams," *IEEE Transactions on Power Delivery,* vol. 18, no. 3, pp. 729-735, 2003.

[25] I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics,* vol. 6, no. 1, pp. 24-43, 2000.

[26] H. Gibson, J. Faith, and P. J. I. v. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," vol. 12, no. 3-4, pp. 324-357, 2013.

[27] Z.-H. Deng and S.-L. J. E. S. w. A. Lv, "Fast mining frequent itemsets using Nodesets," vol. 41, no. 10, pp. 4505-4512, 2014.

[28] E. M. Reingold and J. S. Tilford, "Tidier Drawings of Trees," *IEEE Transactions on Software Engineering,* vol. SE-7, no. 2, pp. 223-228, 1981.