A Modeling Framework to Integrate Exogenous Tools for Identifying Critical Components in Power Systems

Saqib Hasan, Abhishek Dubey, Ajay Chhokra, Nagabhushan Mahadevan, Gabor Karsai and Xenofon Koutsoukos
Institute for Software-Integrated Systems, Vanderbilt University, Nashville, TN 37212, USA
Email:{saqibhasan,dabhishe,chhokraad,nag,gabor}@isis.vanderbilt.edu

***Abstract*—Cascading failures in electrical power systems are one of the major causes of concern for the modern society as it results in huge socio-economic loss. Tools for analyzing these failures while considering different aspects of the system are typically very expensive. Thus, researchers tend to use multiple tools to perform various types of analysis on the same system model in order to understand the reasons for these failures in detail. Modeling a simple system in multiple platforms is a tedious, error prone and time consuming process. This paper describes a domain specific modeling language (DSML) for power systems. It identifies and captures the right abstractions for modeling components in different analysis tools. A framework is proposed that deals with system modeling using the developed DSML, identifying the type of analysis to be performed, choosing the appropriate tool(s) needed for the analysis from the tool-chain, transforming the model based on the required specifications of a particular tool and performing the analysis. A case study is done on WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System to demonstrate the entire workflow of the framework in identifying critical components for power systems.**

***Index Terms*—blackouts, cascading failures, cyber-faults, contingency analysis, DSML, model transformation, protection assembly.**

I. Introduction

Cascading failures in electrical power systems are one of the major causes of concern for the modern society as it results in huge socio-economic loss. These failures can occur from multiple causes such as cyber-attacks, protection equipment mis-operation, system overloading, voltage collapse etc. Recent blackouts of Aug 2003 Northeast USA [1], 2003 Italian blackout [2] have shown electric power grid vulnerability due to such causes and provided reasons to look deeply into the possible sources for these failures. Detailed understanding of cascading failures and identifying critical components for improving system reliability and resiliency necessitates the need to include different aspects (such as steady state vs transient analysis, time independent vs time based analysis, considering protection assembly failures etc.) of the system while performing cascading failure studies. Platforms including various aspects of the system either do not exist or are typically very expensive. Therefore, researchers tend to use multiple open source tools, which are easily available to perform disparate types of analysis on individual platforms. However, these tools have their own specifications and semantics for system modeling and are limited in their capabilities.

Comprehensive understanding for system failure requires modeling of a system in multiple tools for in-depth analysis. For instance, OpenDSS [3], an open source (steady state analysis) tool for electrical power systems can be easily used for quickly identifying critical components based on initial line outages resulting in overloads. However, it is time independent and does not include the modeling of protection assemblies in its simulation environment. These aspects are important while studying cascading failures in detail, as any random outage can change the entire course of the cascade evolution path and can cause severe outages. Researchers interested in analyzing such failures will ultimately look for other simulation platforms (such as Matlab/Simscape [4]) with these capabilities to perform the desired analysis by modeling the same system in it. Moreover, they sometimes use multiple platforms to validate their analysis results.

System modeling in multiple simulation platforms is a tedious, error prone and time consuming process. For e.g., it takes $\sim$ 2-3 hours to model the IEEE-14 Bus System [5] in OpenDSS (including the calculations needed to be done before modeling) and $\sim$ 5-6 hours to model it in Simscape. Considering this, one can only imagine the complexity of modeling systems on a large scale in different platforms. This necessitates the need for a domain specific modeling language (DSML) which can provide the capability to capture the right abstractions for the modeling components of individual low level modeling and simulation tools in a single higher level modeling and simulation platform. System modeling errors and modeling time can be greatly reduced as this DSML is a common language from where other models can be derived.

Prior approaches for cascading failure analysis are based on determining the current state of power system and then to study its evolution using different cascade simulation models [6]–[11]. These approaches can be performed using time independent platforms such as OpenDSS. While it is ideal to use such a platform for expeditious and uncomplicated analysis but performing an in-depth analysis, considering other factors such as time and protection assembly failures due to cyber-faults etc. requires system analysis in a different platform such as Simscape. This facilitates a dynamic analysis providing an advantage over the above models and helps in finding more critical components by employing a richer analysis (not possible otherwise).

A large number of modeling languages are currently available. Modelica [12] is a multi domain modeling language and both commercial and free Modelica simulation environments such as Dymola [13], MapleSim [14] and OpenModelica [15] are available. InterPSS (AC loadflow analysis) [16], PSAT (continuation and optimal power flow) [17], VST (continuation power flow, voltage stability analysis) [18], MATPOWER (optimal power flow) [19] are some of the modeling and simulation tools for cyber physical energy systems for generation, transmission and distribution. Another modeling, simulation and analysis tool for these systems is GridLAB-D [20] and the modeling language is known as GLM. PowerFactory [21] and PSCAD [22] are some of the conventional standard solutions for simulation and analysis purposes. PowerFactory can perform both AC and DC load flow analysis. However, PSCAD is a transient simulation engine. All these modeling languages and tools provide the capability to model the system in their own specific environments with precise input data formats and can perform analysis only based on their individual capabilities. However, most of them do not provide the ability to transform models into a different platform if needed taking into account distinct input data formats and perform the analysis based on the potentials of other tools.

This paper utilizes the concepts of model integrated computing (MIC [23]) to describe a domain specific modeling language for power systems using WebGME (Web-based Generic Modeling Environment) [24], [25]. It identifies and captures the right abstractions for modeling components in different simulation tools (OpenDSS and Matlab/Simscape). A framework is proposed that deals with system modeling using the developed DSML, identifying the type of analysis to be performed, choosing the appropriate tool(s) needed for a particular analysis from the tool-chain, transforming the model(s) based on the required specifications of a particular tool and performing the analysis. Transformed models and supporting executables are generated in order to save system modeling time and to ease the analysis process in multiple platforms. Type checking is also employed to minimize human errors during system modeling. Modeling abstraction is validated using the transformed models of the standard WSCC-9 Bus System [26]. Since the focus of this paper is to identify critical components in electrical power systems, a case study is done on WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System [27] to demonstrate the entire workflow of the framework in identifying critical components.

The paper is organized as follows: Section II describes the modeling language. Section III discusses the system framework. Model transformation and validation is explained in Section IV. The results are demonstrated in Section V followed by the conclusion in Section VI.

II. Modeling Language

A domain-specific modeling language (DSML) has been developed for cyber-physical energy systems (CPS) to enable the rapid design, development and analysis on electrical power systems. A DSML is a declarative language that uses appropriate notations and abstractions to represent various facets of a system and is usually restricted to a particular domain, e.g., power systems.

The meta-model is encapsulated from the developer mode of the graphical interface (WebGME) for model specification, which allows viewing, modification and specification of the rules that administer the construction of power system models and is shown in Figure 1. Every object has a *name* attribute of type *string* and objects with a grayed-out name and in italics is a pure *abstract* object. These *abstract* objects cannot be instantiated in a model but they rather serve as the base class for other instantiable classes. The modeling language captures

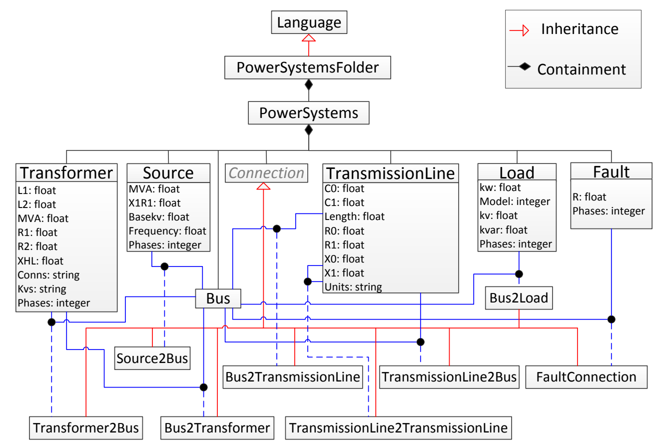

Fig. 1: Modeling Language- UML Class Diagram.

most of the relevant aspects of an electrical power transmission system and using this language engineers can create models containing instances of the objects defined in the DSML. This approach to define the semantics of the models enables a check and ensures model correctness and provides the ability to develop generic utilities called *plugins*. These *plugins* can act on the models created using the modeling language and perform different tasks, for instance model transformation as per the requirements of a particular platform (OpenDSS or Simscape), perform the desired analysis and manage the results. It also supports code development to perform tasks as per user requirements. Although the meta-model captures many aspects of the electrical power systems but it is not a gold-standard model. It is a result of the development, deployment and analysis experiences with different tools. However, the ability to specify the meta-model and build models based on it within the same tool (WebGME) allow the users to extend or modify the meta-model based on their needs.

Figure 1 shows the meta-model as a UML class diagram [28] of the modeling language for power systems. *PowerSystemsFolder* is inherited from *Language*. This *PowerSystemsFolder* contains one or more *PowerSystems*. These *PowerSystems* are the models created using the developed DSML. *PowerSystems* contain one or more *Sources*, *Buses*, *Transformers*, *TransmissionLines*, *Loads*, *Faults* and *Connections*. Each of these objects has a set of attributes that define their individual properties. For instance, the *Source* object has attributes that define its output power, internal source impedance, basekv, frequency of the source voltage and current, number of phases for a source. The attributes are associated with a data type, thereby enabling automatic type checking. These objects are connected together using the rules defined by the *Connection* object. *Connection* are of various types namely *Source2Bus*, *Transformer2Bus*, *Bus2Transformer*, *Bus2TransmissionLine*, *TransmissionLine2TransmissionLine*, *TransmissionLine2Bus*, *Bus2Load* and *FaultConnection*. To ensure model correctness specific *Connection* objects are used. For instance, a *Source* can be connected to a *Bus* using a *Source2Bus* connection but it cannot be connected to a *TransmissionLine* without a *Source2Bus* and *Bus2TransmissionLine* connection. The connectivity of different objects using *Connection* object is shown by solid and dotted blue lines in Figure 1. Once different objects are connected together a power system model is created and made available for analysis purposes.

III. System Framework

The proposed framework enables us to develop domain-specific modeling language (DSML) for power systems. It allows model building depending upon the semantics and rules defined in the modeling language and minimizes modeling errors through type checking. Models are transformed to different simulation platforms considering their individual specifications or input data formats thereby greatly reducing system modeling time and effort. Moreover, it identifies appropriate tool(s) from the tool-chain to perform the desired analysis on the system and manages the results post analysis.

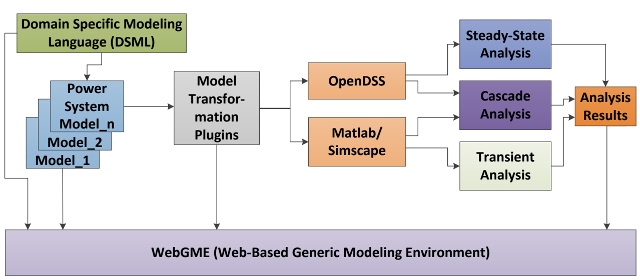

Fig. 2: System Framework

Figure 2 demonstrates the proposed framework where an extendable and specialized tool WebGME is used to orchestrate the workflow. Using the developer mode of this tool a domain-specific modeling language is developed once. This language is used to create power system models within the WebGME modeling environment. Once these models are built they can be transformed based on the requirements of different simulation tool(s), for example OpenDSS, Matlab/Simscape using the model transformation plugins which are specifically developed for model transformation. As WebGME is extendable, these dedicated plugins constitute the supporting infrastructure of the framework. Furthermore, these plugins also perform type checking on the models to ensure correct transformation, for instance they do not let duplicate named objects to be created during the transformation phase which will result in an erroneous model for the simulation tools. Other modeling transformations are also implicitly taken care of during this phase. After the model transformation, appropriate simulation tool is identified from the tool-chain and the model is automatically simulated based on the type of analysis required on the transformed model(s). The type of analysis will depend upon the needs and requirements of a user, for instance steady-state analysis, transient analysis, cascade analysis etc. Finally, post analysis results are gathered back at the WebGME environment. These results can be processed in multiple ways as WebGME is capable of facilitating graphical visualizations as well.

IV. Model Transformation and Validation

Model transformation provides the capability of transforming the model(s) built in WebGME using the developed DSML into the required platform(s) by taking into account the modeling semantics and specifications of individual platform. It ensures model correctness and greatly reduces the time and effort for system modeling in multiple platforms. OpenDSS and Simscape are the two tools used in this framework. Model transformation is performed on WSCC-9 Bus System created in WebGME using the DSML to the models that comply with the modeling semantics of the two tools.

A. *WSCC-9 Bus System WebGME Model*

Domain-specific modeling language discussed in Section II is used to model the WSCC-9 Bus system in the modeling environment of WebGME and is shown in Figure 3. Objects

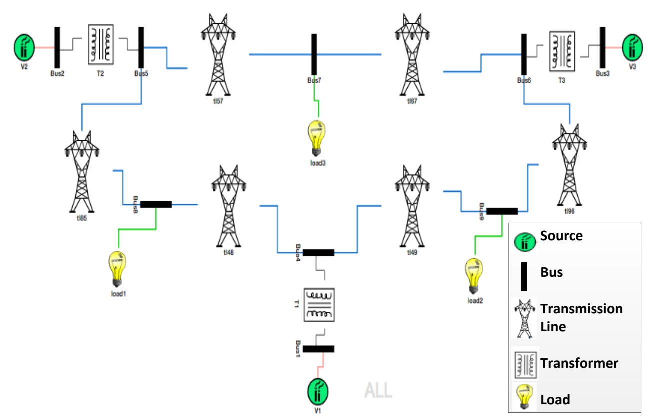

Fig. 3: WSCC-9 Bus System WebGME Model

such as *Sources*, *TransmissionLines*, *Buses*, *Transformes* and *Loads* are selected to model the system. Attributes associated to each object are set with the appropriate data obtained from the IEEE common data format as referenced in [26].

B. *WSCC-9 Bus System OpenDSS Model*

OpenDSS is a time independent, script based steady-state power system modeling and simulation tool. The WSCC-9 Bus System WebGME model is automatically transformed to the OpenDSS model using dedicated *plugins* constituting the framework discussed in Section IV. As shown in Figure 4, the model is transformed by taking into account every object (*Sources, Buses* etc.) and its associated attributes to the appropriate semantics in OpenDSS. During the transformation, type checking is employed to ensure proper data flow for each object and to identify and remove duplicate object names which can cause compilation error during model simulation.

```
clear
New object=circuit.9bus
//Define Sources
New vsource.Source1 bus1=Bus1 phases=3 basekv=16.5 Mvasc3=247.5 r1=.0000001 x1=0.0000001
New vsource.Source2 bus1=Bus2 phases=3 basekv=18 Mvasc3=192 r1=.0000001 x1=.0000001
New vsource.Source3 bus1=Bus3 phases=3 basekv=13.8 Mvasc3=128 r1=.0000001 x1=.0000001
//Define the transmission lines and transformers
New Line.TL48 bus1=Bus4 bus2=Bus8 R1= 0.0529 R0=0.13225 X1=.4494 X0=.8972 C1=8.82  C0=5.188 length=62.1371 units=mi
New Line.TL49 bus1=Bus4 bus2=Bus9 R1=0.08993 R0=0.224825 X1=.4863 X0=1.2139 C1=7.922 C0=4.74 length=62.1371 units=mi
New Line.TL85 bus1=Bus8 bus2=Bus5 R1=0.16928 R0=0.4232 X1=.8516 X0=2.1262 C1=15.34 C0= 9.025 length=31.0686 units=mi
New Line.TL96 bus1=Bus9 bus2=Bus6 R1=0.20631 R0=0.5157 X1=.8972 X0=2.2959 C1=17.95 C0= 10.55 length=62.1371 units=mi
New Line.TL57 bus1=Bus5 bus2=Bus7 R1=0.044965 R0= 0.11241 X1=.3808 X0=.7615 C1=7.471 C0= 4.394 length=62.1371 units=mi
New Line.TL67 bus1=Bus6 bus2=Bus7 R1=0.062951 R0= 0.15737 X1=.5331 X0=1.3308 C1=10.47 C0= 6.15 length=62.1371 units=mi
New transformer.T1 phases= 3 buses= (Bus1 Bus4) Kvas=[100000 100000] conns= 'wye wye' kvs= "16.5 230" XHL=5.7147
New transformer.T2 phases= 3 buses= (Bus2 Bus5) Kvas=[100000 100000] conns= 'wye wye' kvs= "18 230" XHL=6.5619
New transformer.T3 phases= 3 buses= (Bus3 Bus6) Kvas=[100000 100000] conns= 'wye wye' kvs= "13.8 230" XHL=5.0917
//Define the loads
New Load.Load1 bus1=Bus8 phases=3 kVA=125000, 50000 Kv=230 conn= delta model=1
New Load.Load2 bus1=Bus9 phases=3 KVA=90000, 30000 Kv=230 conn= delta model=1
New Load.Load3 bus1=Bus7 phases=3 kVA=100000, 35000 Kv=230 conn= delta model=1
//Define the voltagebases
set voltagebases=[16.5, 18, 13.8, 230]
calcv
set freq=60
set mode=snapshot
solve
```

Fig. 4: WSCC-9 Bus System OpenDSS Model

C. WSCC-9 Bus System Matlab/Simscape Model

Matlab is a time-based modeling and simulation tool. It has the capability to extend itself and perform the necessary simulation and analysis based on the users needs and requirements. Moreover, Matlab can be easily used for transient analysis in electrical power systems. The WebGME model for the WSCC-9 Bus System is automatically transformed to the appropriate simulation model using dedicated *plugins* constituting the framework discussed in Section IV and is shown in Figure 5. The transformed model takes into account

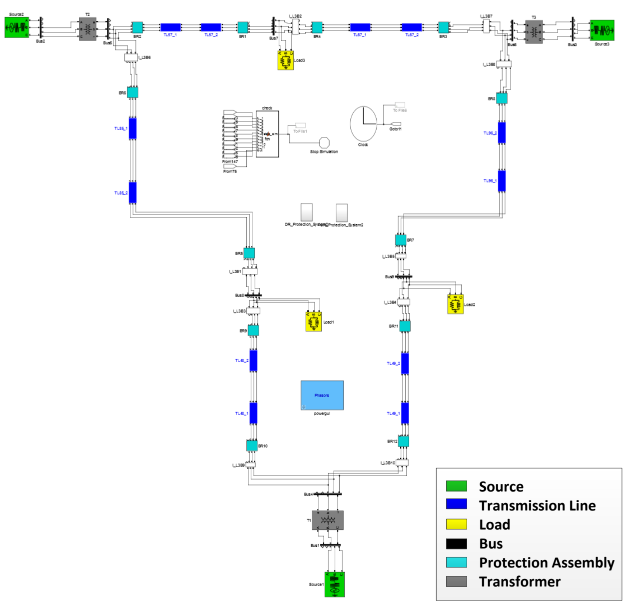

Fig. 5: WSCC-9 Bus System Matlab/Simscape Model

every object and its associated attributes to represent the model with correct semantics of Simscape. Certain object attributes require conversion while modeling system in multiple platforms. For e.g., the data obtained from IEEE common data format for the transmission lines in WSCC-9 Bus System has line reactances (X) but it does not contain line inductances (L), which needs conversion using the formula L = X/(2*pi*f). Such conversions are automatically taken into account using the model transformation plugins as OpenDSS model takes line reactances (X) as inputs to its transmission line objects and Simscape model takes line inductances (L) as inputs for its transmission line model block.

Furthermore, model transformation saves a lot of time and effort. For instance, the model transformation plugins for Simscape model automatically inserts the current and voltage measurement blocks and a protection assembly block at each end of a transmission line. These blocks are needed for the analysis but are not defined as objects to reduce the complexity and to give a higher abstraction to the DSML. The protection assembly blocks are custom designed and pre-added to the Simscape library to facilitate the model transformation process. These blocks provide the capability to introduce cyber-faults in addition to the physical faults in electrical power systems at different instants. Details about the behavior models of protection assembly blocks considering cyber-faults is referenced in [29].

D. Validation of The Transformed Models

To Validate the transformed models for the WSCC-9 Bus System, direct mapping of the objects from DSML to OpenDSS and Matlab/Simscape are listed in Table I.

TABLE I: DSML Object mapping to OpenDSS and Simscape.

DSML Object Name	**OpenDSS Object Name**	**Matlab/Simscape Block Name**
Source	Vsource	Three-Phase Source
TransmissionLine	Line	Three-Phase PI Section Line
Transformer	Transformer	Three-Phase Transformer (Two Windings)
Bus	Bus	Three-Phase VI Measurement
Load	Load	Three-Phase Parallel RLC Load
Fault	Fault	Three-Phase Fault

The transformed models of WSCC-9 Bus System are simulated in the two platforms (OpenDSS, Matlab/Simscape) under nominal mode (absence of any fault condition). These models yield the same numerical values of bus voltages and transmission line currents with an average error of $\sim$ 1% for bus voltages and $\sim$ 3% for line currents. This variation is attributed to the different solvers in the two platforms.

V. Results

Using the framework discussed in Section III, critical components causing cascading failures resulting in blackouts are identified using the cascade analysis performed on OpenDSS and Matlab/Simscape models of WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System. Here, blackout criteria is considered as 40% of system load loss which is one of the criterion referenced in [30] and transmission lines are assumed to be loaded at 70% of their loading capacity for each system. Cascading analysis due to initial line outages resulting in subsequent components overloading are performed using OpenDSS (quick and easy, time-independent analysis). However, time based cascade analysis due to physical faults in transmission lines (for instance 3-phase to ground fault) and cyber-faults in protection assemblies are performed using Matlab/Simscape. Details about modeling cyber-faults in protection assembly and their integration with the Matlab/Simscape models to perform cascade analysis are presented in [29].

A. *OpenDSS-Time Independent Analysis*

The transformed models of WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System created using the developed DSML are used to perform the time-independent cascade analysis to identify critical components (transmission lines) causing blackout. As OpenDSS do not have an object to define protection assembly (distance relay, over-current relay and circuit breakers) and the cyber-faults associated with it, these models cannot be used to perform detailed analysis to identify critical protection assemblies causing blackout. Although, behavior of some cyber-faults in protection assemblies can be replicated in OpenDSS but it requires manually changing the OpenDSS model which is a very tedious process. Moreover, timing information which is useful for the operators cannot be obtained using this analysis. However, it serves as an ideal way to quickly identify critical transmission lines based on line overloading.

A simple cascade analysis framework is implemented using the COM interface in OpenDSS. N-k (N = No. of components, $k \in \mathbb{N}$) contingency analysis is performed to identify critical components based on initial line outages. These outages are a set of combinations of line outages that are iteratively removed from the network to simulate the system for possible blackouts. For instance, if k=2 then the set of initial line outages will have a total number of $\binom{N}{2}$ combinations. Each combination

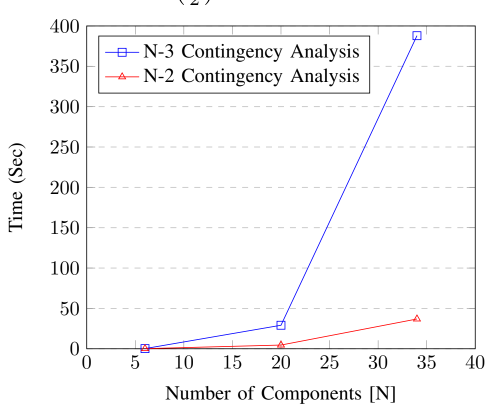

Fig. 6: Contingency Analysis

of outage(s) are tripped from the network and the system is checked for overloads. If it exits, the overloaded branches (transmission lines and transformers) are tripped. The system is checked for blackout criteria and if it is met the simulation is stopped and the initial outage(s) are marked as critical. If the blackout criteria is not met and there are further overloads the required branches are tripped and the check is performed again. If there are no further overloads and blackout criteria is not met then the initial outage(s) are not classified as critical. More detailed explanation is referenced in [29].

As per NERC standards, power systems are N-1 tolerant, hence N-2 and N-3 contingency analysis is performed on each system to identify combinations of critical transmission lines causing blackout. Based on the above cascade analysis framework, for N-2 contingency analysis, a total of 168 (13+40+115) combinations out of 901 (15+190+561) combinations of line outages were observed to cause blackout in WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System respectively. For N-3 contingency analysis, a total of 2515 (20+400+2095) combinations out of 7144 (20+1140+5984) combinations of line outages were observed to cause blackouts in the above mentioned systems. These combinations are marked as critical lines and can help in improving system resiliency. Figure 6 shows the plot of time taken to run the analysis for each system versus the number of components in each system. As 'k' increases the analysis time increases more with increase in the number of components and the plot becomes more exponential. However, this may not be an issue as it is an off-line analysis and does not take a significant amount of time. This can further be improved by employing parallel computing.

B. *Matlab/Simscape-Time based Analysis*

Transformed models of WSCC-9 Bus System, IEEE-14 Bus System and IEEE-39 Bus System are used to perform the time-based cascade analysis but only the results of IEEE-14 Bus System are shown due to space constraints. In this analysis

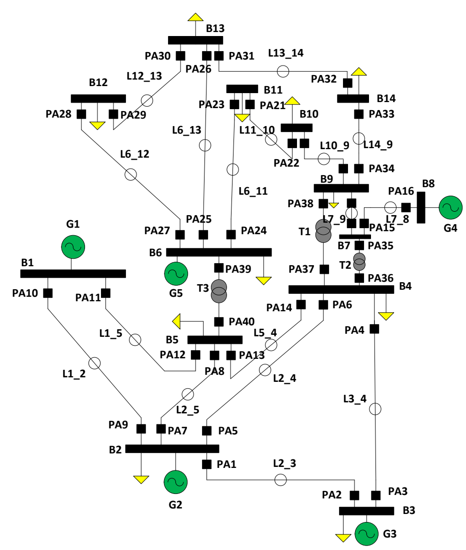

Fig. 7: IEEE-14 Bus System [5]

cyber-faults in protection assembly (details about cyber-faults and its modeling in the protection assembly is referenced in [29]) causing cascading failures resulting in blackout are considered and critical protection assemblies are identified. It is a time-based analysis and can be useful for operators to design effective mitigation strategies as details about every failure are available with respect to time.

Analysis is performed on the IEEE-14 Bus System and every transmission line is protected using a pair of protection assembly (represented by PAn, $n \in \mathbb{N}$, as shown in Figure 7). Protection assembly consists of a distance relay, an over-current relay and a circuit breaker (dentoted as PA_DRn, PA_ORn and PA_BRn respectively). Each line is given a physical fault (3-phase to ground fault) and the associated circuit breakers are given a *Stuck Close Breaker Fault* (a type of cyber-fault where the circuit breakers do not operate as desired) individually. Results of other cyber-faults are not shown due to space constraints. The simulation is run using the cascade simulation framework discussed in Section V(A). Initial outages are a combination of physical fault and a cyber-fault (referenced in [29]). As per the blackout criterion, three highly vulnerable protection assembly components (PA_BR4, PA_BR13, PA_BR14) are observed in the system with this fault combination. Based on the study, critical components are identified and categorized in Table II. Components listed in 'Category I' are the components that causes a blackout in the presence of a physical fault and a cyber-fault. However, the components in 'Category II' are likely to cause a blackout if there is any other outage that results in further load loss. These are less critical compared to 'Category I' but still should be considered while improving system resiliency. 'Category III' components are not as critical as the other two categories but can result in blackouts if drastic load loss happens due to a large number of outages.

TABLE II: Critical Components Categorization

Category Name	**Component Name**	**Load Loss**
Category I	PA_BR4, PA_BR13 PA_BR14	above 40%
Category II	PA_BR6, PA_BR7	very close to 40% (39.22%)
Category III	PA_BR18, PA_BR22 PA_BR34	$>$ 25% and $<$ 35%

VI. Conclusion

In this paper, a domain-specific modeling language (DSML) for electrical power systems is described that identifies and captures the right abstractions for modeling components in different simulation tools. A framework is proposed to facilitate the development of DSML, model creation and transformation and to perform the desired analysis by choosing appropriate tool from the tool-chain. A case study is performed on WSCC-9 Bus system, IEEE-14 Bus System and IEEE-39 Bus System to show how this framework is used in identifying critical components in power systems. Moreover, the design provides flexibility to easily understand and extend the DSML and the supporting infrastructure based on the users needs and requirements. It also provides the capability to integrate more simulation tools so as to perform the desired analysis from within a single environment. As part of the future work, more complex models need to be analyzed and the entire approach can be automated, to perform the desired analysis from within the WebGME environment.

Acknowledgment

This work is funded in part by the National Science Foundation under the award number CNS-1329803 and the FORCES project under the award number CNS-1238959.

References

[1] U.-C. Force, "Final report on the august 14th blackout in the united states and canada," *Department of Energy and National Resources Canada*, 2004.
[2] A. Berizzi, "The italian 2003 blackout," in *Power Engineering Society General Meeting, 2004. IEEE*. IEEE, 2004, pp. 1673–1679.
[3] O. P. Model and O. S. Element, "Opendss manual," *EPRI,[Online] Available at: http://sourceforge. net/apps/mediawiki/electricdss/index. php*.
[4] http://www.mathworks.com/, The mathworks tools.
[5] http://icseg.iti.illinois.edu/ieee-14-bus-system/, IEEE-14 Bus System.
[6] I. Dobson, B. A. Carreras, and D. E. Newman, "A loading-dependent model of probabilistic cascading failure," *Probability in the Engineering and Informational Sciences*, vol. 19, no. 01, pp. 15–32, 2005.
[7] J. Chen and J. Thorp, "A reliability study of transmission system protection via a hidden failure dc load flow model," in *Power System Management and Control, 2002. Fifth International Conference on (Conf. Publ. No. 488)*. IET, 2002, pp. 384–389.
[8] P. D. H. Hines, I. Dobson, E. Cotilla-Sanchez, and M. Eppstein, ""dual graph" and "random chemistry" methods for cascading failure analysis," in *Proceedings of the 2013 46th Hawaii International Conference on System Sciences*, ser. HICSS '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 2141–2150.
[9] Y. Koç, T. Verma, N. A. Araujo, and M. Warnier, "Matcasc: A tool to analyse cascading line outages in power grids," in *Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on*. IEEE, 2013.
[10] B. A. Carreras, D. E. Newman, I. Dobson, and N. S. Degala, "Validating opa with wecc data." in *HICSS*, 2013, pp. 2197–2204.
[11] D. P. Nedic, I. Dobson, D. S. Kirschen, B. A. Carreras, and V. E. Lynch, "Criticality in a cascading failure blackout model," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 9, pp. 627–633, 2006.
[12] P. Fritzson, *Principles of object-oriented modeling and simulation with Modelica 2.1*. John Wiley & Sons, 2010.
[13] http://www.3ds.com/, Dymola Tool.
[14] https://www.maplesoft.com/products/maplesim/, MapleSim Tool.
[15] P. Fritzson, P. Aronsson, H. Lundvall, K. Nyström, A. Pop, L. Saldamli, and D. Broman, "The openmodelica modeling, simulation, and software development environment," *Simulation News Europe*, vol. 44, 2005.
[16] M. Zhou and S. Zhou, "Internet, open-source and power system simulation," in *Power Engineering Society General Meeting*. IEEE, 2007.
[17] F. Milano, "An open source power system analysis toolbox," *IEEE Transactions on Power systems*, vol. 20, no. 3, pp. 1199–1206, 2005.
[18] S. Ayasun, C. O. Nwankpa, and H. G. Kwatny, "Voltage stability toolbox for power system education and research," *IEEE Transactions on Education*, vol. 49, no. 4, pp. 432–442, 2006.
[19] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2011.
[20] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, "Gridlab-d: An open-source power systems modeling and simulation environment," in *IEEE/PES Transmission and Distribution Conference and Exposition, 2008*.
[21] http://www.digsilent.de/, PowerFactory Tool.
[22] https://hvdc.ca/pscad/, PSCAD Tool.
[23] J. Sztipanovits and G. Karsai, "Model-integrated computing," *Computer*, vol. 30, no. 4, pp. 110–111, 1997.
[24] https://webgme.org/, The WebGME tool.
[25] Z. Lattmann, T. Kecskés, P. Meijer, G. Karsai, P. Völgyesi, and Á. Lédeczi, "Abstractions for modeling complex systems," in *International Symposium on Leveraging Applications of Formal Methods*. Springer, 2016, pp. 68–79.
[26] http://icseg.iti.illinois.edu/wscc-9-bus-system/, WSCC-9 Bus System.
[27] http://icseg.iti.illinois.edu/ieee-39-bus-system/, IEEE-39 Bus System.
[28] G. Booch, *The unified modeling language user guide*, 2005.
[29] S. Hasan, A. Chhokra, N. Mahadevan, A. Dubey, and G. Karsai, "Cyber-physical vulnerability analysis," no. ISIS-17-101, pp. 1–14, 2017.
[30] http://sites.ieee.org/pes-cascading/presentations/, PES General Meeting.