Bus Inf Syst Eng 65(6):643–676 (2023)
https://doi.org/10.1007/s12599-023-00811-0

RESEARCH PAPER

Model-Based Cybersecurity Analysis

Extending Enterprise Modeling to Critical Infrastructure Cybersecurity

Yuning Jiang · Manfred A. Jeusfeld · Jianguo Ding · Elin Sandahl

Received: 26 July 2021 / Accepted: 20 January 2023 / Published online: 6 May 2023

Abstract Critical infrastructure (CIs) such as power grids link a plethora of physical components from many different vendors to the software systems that control them. These systems are constantly threatened by sophisticated cyber attacks. The need to improve the cybersecurity of such CIs, through holistic system modeling and vulnerability analysis, cannot be overstated. This is challenging since a CI incorporates complex data from multiple interconnected physical and computation systems. Meanwhile, exploiting vulnerabilities in different information technology (IT) and operational technology (OT) systems leads to various cascading effects due to interconnections between systems. The paper investigates the use of a comprehensive taxonomy to model such interconnections and the implied dependencies within complex CIs, bridging the knowledge gap between IT security and OT security. The complexity of CI dependence analysis is harnessed by partitioning complicated dependencies into cyber and cyber-physical functional dependencies. These defined functional dependencies further support cascade modeling for vulnerability severity assessment and identification of critical components in a complex system. On top of the proposed taxonomy, the paper further suggests power-grid reference models that enhance the reproducibility and applicability of the proposed method. The methodology followed was design science research (DSR) to support the designing and validation of the proposed artifacts. More specifically, the structural, functional adequacy, compatibility, and coverage characteristics of the proposed artifacts are evaluated through a three-fold validation (two case studies and expert interviews). The first study uses two instantiated power-grid models extracted from existing architectures and frameworks like the IEC 62351 series. The second study involves a real-world municipal power grid.

Keywords Critical infrastructure · Domain-specific language · Cybersecurity · Power grids

1 Introduction and Background

Critical infrastructure systems (CIs), such as energy and water distribution, and transportation roadways, are vital to maintaining the normalcy of society (Humayed et al. 2017). CI typically combines information technology (IT) and operational technology (OT) systems that are converging due to the drive towards data-driven and remote operations (Murray et al. 2017). Meanwhile, the rapid advances in information and communication technology (ICT) enable seamless integration of software and hardware, towards a shift from diverse systems empowered mainly by either hardware or software to cyber-physical systems (CPSs) driving emergent systems including Industry 4.0 evolution (Alcaraz 2019; Xu et al. 2018).

Accepted after 2 revisions by Hajo Reijers.

Y. Jiang: most of the work by the contact author was done while at the University of Skövde.

Y. Jiang (✉)
Nanyang Technological University, Singapore 639798, Singapore
e-mail: yuning.jiang@ntu.edu.sg

Y. Jiang · M. A. Jeusfeld
University of Skövde, 541 28 Skövde, Sweden

J. Ding
Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

E. Sandahl
Norgald AB, Långlandia 2B, 411 33 Göteborg, Sweden

However, alongside the expected enhancement in efficiency and reliability, the induced connectivity prompted by ICT and its application in Supervisory Control and Data Acquisition (SCADA) systems expose these CIs to cyber-attacks where conventional security approaches are limited by the scale of the infrastructures (He and Yan 2016; Nguyen et al. 2017; Cheminod et al. 2012). Some well-known attacks demonstrate these threats to CIs, like the Stuxnet worm (Falliere et al. 2011) and the "WannaCry" ransomware (Mohurle and Patil 2017). Stuxnet was first encountered in 2009 and did not raise broad discussions until 2010. In 2017, the "WannaCry" ransomware attack occurred across several CIs and caused production to stop, incurring substantial business losses. Furthermore, traditional IT attack methods such as credential theft and DoS are proving to be just as effective on OT networks (Bhamare et al. 2020). Attackers often start on the IT network and use IT assets as jump servers to move to more critical OT assets, which results in a severe impact on CIs. For example, the Ukraine power grid attack in 2015 (Whitehead et al. 2017) is a known attack against the power grid system while directly targeting the OT system.

1.1 Background

An approach to address dependencies and vulnerabilities across components enables an online collection of relevant data to assess vulnerability properties in CIs and adopt proper defense mechanisms. However, CI-related data is massive with a significant level of heterogeneity that needs to be transformed into a common semantic representation to facilitate machine-readable processes, in order to improve situation awareness applications.

For complex CIs, the layered approach helps to understand the operating logic of the system and related network physical control functions (Mo et al. 2011). A power grid is a typical CI distinguished by the enormous scale and intricate interconnections of the network carrying power flows. The network includes power components tied up together via transmission and distribution lines to form a complex system connecting power-generation sources to power-consuming loads. Smart power grid employs CPS in evolution from aging power-delivery systems to optimize and protect electricity delivery operations (Humayed et al. 2017). These processes could be facilitated by analyzing data from the different layers composing the power-grid architectures. For instance, the control layer includes a network of microprocessor-controlled physical objects, such as remote terminal units (RTUs), which interface with physical process sensors and actuators. On top of the cyber-physical layer, control center applications process these measurements to support operational power-flow decisions to balance the supplied and demanded power flows (Knapp and Langill 2014). Figure 1 illustrates the interplay across evolving CI layers, which exhibit the complex dependencies of such systems, especially between cyber and physical layers. Such a new interdependent relationship between different systems also introduces new vulnerabilities to the system. And hence, these dependencies need to be identified and assessed systematically.

1.2 Problem Statement

With the introduction of modern smart grid functionality, the increasing share of regionally produced renewable energy and the current shift to electric vehicles, the number of computer-controlled components is increasing. At the same time, the control infrastructure is increasingly decentralized, and knowledge about dependencies between components is decreasing. Improving the cybersecurity of CIs for both IT and OT networks is vital. However, OT security, especially the security of CPS-based field devices, is overlooked by cybersecurity professionals partially due to the "air-gaped"operation isolation of previous OT devices (Murray et al. 2017). The aforementioned trend of IT and OT convergence exposes relatively isolated OT equipment to the risks common in IT security protection. Nevertheless, IT and OT cybersecurity practices normally have different priorities, meaning IT security usually focuses on the confidentiality, integrity, and availability of critical data, while OT security concerns more on the protection of production loss or safety (Conklin 2016). Limited collaboration between different departments of IT and OT also contributes to the knowledge gaps in cybersecurity assessment of CIs (Vielberth et al. 2020). Meanwhile, it is challenging to extract and manage system configuration information from CIs (Bernstein and Haas 2008). Normally, operators need to query different PCs/machines following various vendors' suggestions. For example, one may obtain embedded software in a Windows computer by using PowerShell (Shepard 2015). However, different vendors utilize various semantics and syntactic, which increases the difficulty in information integration. Second, many critical infrastructure companies outsource their IT or OT services to other companies, which further enlarges the knowledge gap between different sub-systems (Kandias et al. 2011). Hence, we need a modeling framework that is extensible in terms of component types and their interconnections.

Model-based approaches have been used to design software systems but also to understand socio-technical and CPSs. Notably, enterprise modeling frameworks such as Aris Architecture (Scheer and Nüttgens 2000) and ArchiMate (Lankhorst et al. 2010) aim at creating integrated models of the business and IT levels of an enterprise and to establish their dependencies. ArchiMate subdivides the

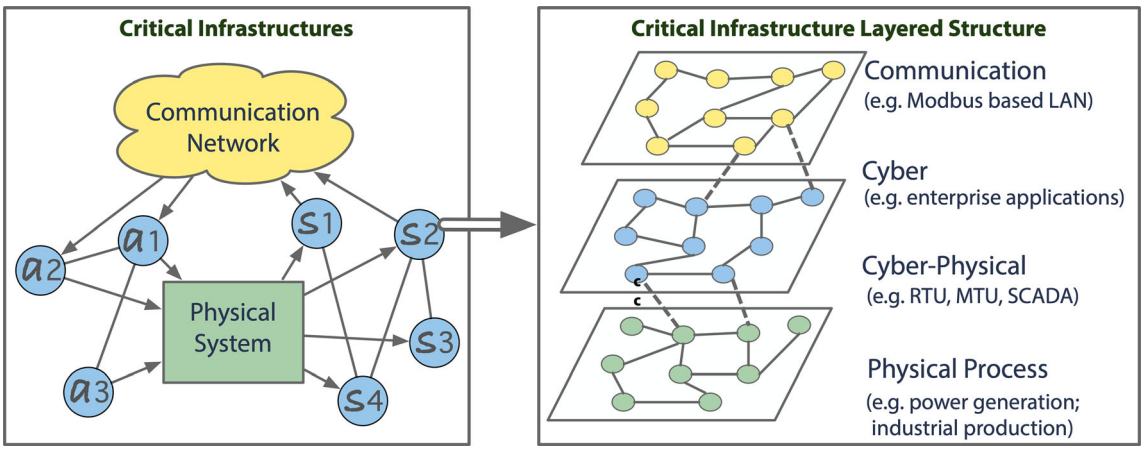


Fig. 1 Critical infrastructure layered architecture

enterprise models into three main layers. The *business layer* covers business processes, business goals, actors, roles and so forth. The *application layer* includes the application software models, their services, and interfaces. Finally, the *technology layer* models system software such as operating systems and database management systems plus the physical components such as computers and network devices. Of particular importance are the explicit dependencies that cross layers. In Aris, these are mostly "implementation"dependencies, e.g., a task in a business process is implemented by application software. ArchiMate has similar dependency link types called "usage"links, which serves the same purpose. The dependency links allow for the analysis of the enterprise models. For example, one can retrieve those business processes that depend on the availability of a given database server at the technology level.

Enterprise modeling has been proposed for supporting cybersecurity management by a number of authors. Janulevičius et al. (2017) use an ontology to describe the enterprise architecture artifacts related to cloud computing in order to decorate the enterprise architecture model with security-related concepts. Specifically, they cover the security aspects of the governance, virtualization, and operation of cloud services. The purpose of this ontology is to guide the enterprise architecture design. Likewise, Pavleska et al. (2019) developed a reference architecture for evaluating the information security of an enterprise architecture. Their conceptual framework covers security goals, vulnerabilities, threats, and security measures, all being linked to the enterprise model. The reference architecture is a guideline to manually assess the security status of an enterprise, using its enterprise model. Burkett (2012) uses the Sherwood Applied Business Security Architecture (SABSA) to incorporate information security considerations into an enterprise architecture. The approach is compatible with popular enterprise architecture frameworks such as the Zachman framework and TOGAF. Rather than proposing a new framework, they propose to augment existing enterprise architecture frameworks with security aspects. A formal enterprise modeling language is not used. Grandry et al. (2013) extend the ArchiMate meta-model by cybersecurity concepts such as risks, threats, vulnerabilities, security goals, and countermeasures. Ekstedt and Sommestad (2009) focus on modeling attack and defense trees as formal artifacts in the meta-model for enterprise architectures. The approach targets critical infrastructure operators, such as power grid companies. The approach has been further elaborated by Sommestad et al. (2013) into the cybersecurity modeling language CySeMoL. It covers typical assets of an IT system such as networks, software products, data flows, data stores, protocols, etc. Leune and Kim (2021) put services in the center of their enterprise modeling tool. Services are defined in terms of their provider, the data flows between them, and flow channels. The implementation is based on ConceptBase and uses the query capabilities of ConceptBase to analyze vulnerabilities of a given enterprise model. Diefenbach et al. (2019) present a systematic literature review on information security integration into enterprise architectures. One finding is that enterprise architecture management is already contributing to improving risk and information security management. However, they argue that more research is needed to properly integrate information security and risk management concepts into enterprise architectures. Mozzaquatro et al. (2016) propose an IoTSec ontology-based framework that combines both model-driven development and ontology-driven development. Their framework covers two use-case scenarios, i.e., one for design purposes and the other for run-time system security monitoring and management. The same authors (Mozzaquatro et al. 2018) extend the IoTSec reference

ontology into a database of IoT cybersecurity knowledge (vulnerability, threat, and prevention mechanism) to support cybersecurity analysis.

There is still a need for a unified ontology for IT/OT security in CIs, as different descriptive terms reported in these two domains bring high heterogeneity and low interoperability (Mohamed et al. 2021). This paper aims to answer the question of how to model CIs to allow vulnerability assessment, considering the cyber-physical interconnections and dependencies within CIs.

1.3 Contribution

This paper proposes an extensible taxonomy that models common semantics of both IT and OT entities to support IT/OT security convergence, which is the continued work of the authors (reference hidden). CI entities and their types and attributes (such as security attributes and dependencies) are defined in this taxonomy. A variant of CI component properties can be further specified, added or removed in the format of attributes, which allows further extensibility. The proposed taxonomy supports CI dependence analysis. Only the dependencies within one infrastructure are modeled. More specifically, we focus on intra-dependencies of CIs, especially in the cyber and cyber-physical domains. However, possible dependencies between different infrastructures are out of scope.

We set up references models for one CI (here, the smart grid) to evaluate the applicability of the proposed taxonomy. We conduct an in-depth literature review to collect information and summarize smart power-grid systems' common topological structure and functional architecture. Power-grid networks are then instantiated on top of our taxonomy to reflect real infrastructure connections and support vulnerability-centered simulations with reliable predictions based on the collected information. This way, the taxonomy and reference model play an important explanatory role in exploring system-wide vulnerabilities due to cyber and cyber-physical dependencies. While we present in this paper only the details and case studies of power grids, the taxonomy can be extended to cover other CIs such as district heating systems. The common denominator of these systems is that there is a physical matter-energy flow that is controlled by information technology.

We further propose a vulnerability assessment method to connect the proposed taxonomy with security repositories such as NIST (2022), or NVD, to identify and assess matching vulnerabilities for CIs. In doing so, individual vulnerabilities and also chained vulnerabilities are analyzed, to prevent advanced persistent threats (APT) (Chen et al. 2014). While the overall approach allows for dynamic simulation, particularly power flow simulation and cyber-attack simulation, we focus on the static analysis of the CI model in this paper.

We implement the proposed taxonomy, instantiated reference models and dependence-analysis deductive rules in a tool named ConceptBase. We published our instantiated models and code (reference hidden). Partial models can be derived from the integrated CI model to create cyber-attack simulation and power flow simulation models. Details are provided in a technical report (reference hidden). An interface to a cyber vulnerability repository has also been realized and tested in a real-world data center located in Sweden (reference hidden). To summarize, the contributions of the paper are as follows:

1. The proposed taxonomy provides a rich set of component and interconnection types to model complex CIs and are instantiated into realistic power-grid systems. This extends the research on enterprise modeling by covering the physical processes below the technology layer of classical enterprise modeling languages. Security of such CIs are addressed by defining vulnerability-centered attributes on top of the Common Vulnerability Reporting Format (CVRF) (Schiffman 2011) framework to allow further security information sharing and enhanced interoperability with other security tools. The process of vulnerability attributes extraction is also clearly clarified.
2. Multiple extensive and realistic reference models were designed to define power-grid system aspects such as the control center, substations, and data/control flows between software components. The reference models validate to suitability of the proposed taxonomy to describe all layers of power-grid systems, from the physical power-grid components to the software applications.
3. The dependence rules are deriving the functional dependence structure from data flow specifications. The rules are fully implemented via the ConceptBase system and efficiently compute the dependencies. The rules can also be used to pinpoint the most "critical"components in a CI model in terms of the number of components that depend on them, see Sect. 5.1 for our proposed metric on direct functional dependence. This extends works on traceability of enterprise models by linking IT components to software artifacts and to the components of the physical grid.

The rest of this paper is organized as follows: In Sect. 2, we review some state-of-the-art of semantic models, reference models, and dependence analysis for CIs, while focusing on power-grid models for better comparisons. Section 3 introduces the adopted design science methodology for our taxonomy design and evaluation. In Sects. 4 and 5, we introduce our taxonomy and its usage in defining

dependencies and vulnerability analysis among CI components, separately. Section 6 presents some reference models for power-grid systems, followed by two case studies and interview result analysis in Sect. 7. The first study queries cyber dependence and conducts a simplified cascade modeling of two instantiated power-grid networks. The second study validates the utility of our model in a real-world power grid. Finally, Sect. 8 concludes our work and shows future directions.

2 Theoretical Background

2.1 Semantic Models and Frameworks for Critical Infrastructure Cybersecurity

Risk assessment of a complex CI such as a power-grid system involves analyzing various vulnerabilities across highly interdependent IT and OT components. Identifying only individual vulnerabilities and threats is not sufficient in today's complex systems (Kure et al. 2018). Different modeling attempts have been made to pinpoint both individual vulnerabilities (e.g., legacy software) and structural vulnerabilities (e.g., lack of network segmentation) (Blockley et al. 2002). Data visualization models like tree structures, directed graphs, and logic diagrams are widely used for system-wide cybersecurity assessment or exploitation modeling (Noel et al. 2016; Lallie et al. 2018). Numerous prior studies based on tree structures or graphs are typically tailored to particular system structures or network environments and assess the probability or potential impact of exploiting certain vulnerabilities such as Denial of Service (DoS) and Man-in-the-Middle (MiTM). Flexibility and extensibility usually are not the prime designing criteria (Noel et al. 2016). In other words, existing cybersecurity assessment frameworks may require substantial reconstruction to validate a different type of vulnerability, and are therefore neither effective nor economical. Moreover, the experimental datasets or evaluation datasets are mostly not published, which makes the process hard to reproduce (Eckhart and Ekelhart 2018).

Many valuable frameworks have been proposed to address the rising security issues in the OT systems, including NIST SP 800-82 for the industrial control system (ICS) security (Stouffer et al. 2011), NIST cybersecurity Framework for Critical Infrastructure (NIST 2014), and NERC (2008) standards. In addition, several international standards specifically focus on security in the domain of smart grid such as IEC 62351 (entitled "*Power systems management and associated information exchange - Data and communications security*") and NISTIR 7628 Rev. 1 (entitled "*Guidelines for Smart Grid cybersecurity*"), that are summarised by Ruland et al. (2017). Due to the IT and OT convergence introduced in the previous section, the scientific community and industry continue to search for solutions to bridge the gaps between IT and OT security. However, Conklin (2016) suggests that the adaption of IT-specific security regulations (e.g., NIST SP 800-53) to OT security directives (e.g., NIST SP 800-82) leaves the fundamental business objective differences between IT and OT systems unaddressed.

Risk modeling languages (e.g., semantic maps and ontologies) for model-based security engineering have been proven to be scalable and flexible (Nguyen et al. 2017; Zhou et al. 2012). Several enterprise architecture frameworks have been developed to support risk presentation and analysis, such as an earlier framework named CORAS by Fredriksen et al. (2002). There are also newer developments like Secure-i* by Liu et al. (2009) and Secure-Tropos by Mouratidis and Giorgini (2007). These semantic ontologies designed for cybersecurity purposes and CI operations are valuable, but need to be merged to achieve an effective cybersecurity analysis (Diefenbach et al. 2019). Recently, (Mohamed et al. 2021) conclude that generic tools like SysML are not really suitable because they do not capture the semantics of the CPS.

Some works (e.g., Venkata et al. (2018); Mozzaquatro et al. (2018)) connect Common Platform Enumeration (CPE) MITRE (2022b) ontology with cybersecurity databases, meaning that the vulnerability information for different components can be integrated into their ontology. In these works, ontologies are applied to provide a formal and explicit way to specify concepts and relationships. In the study by Venkata et al. (2018), for instance, public vulnerability data seeds from repositories like Common Vulnerability Enumeration (CVE) MITRE (2022c) and CPE are correlated to their ontology knowledge base, and further mapped through the STRIDE (Khan et al. 2017) threat categorization. However, this work does not consider further reasoning and logical analysis of how their ontology correlates to various vulnerabilities, threats, and mitigations.

We build our taxonomy upon the existing models and our investigations in CI architectures. For example, we adopted and built on top of the CVRF (Schiffman 2011) when developing vulnerability attributes of our CI entities, to allow more accessible cybersecurity information sharing with major security alert repositories such as NVD. In doing so, we suggest a taxonomy that contains not only both IT/OT components, software installed on these components, but also their properties such as potential vulnerabilities, as well as their interconnections, which can be used as a basis to perform the IT/OT convergence studies. We then instantiate power-grid reference models that show the utility of our taxonomy for bridging different

terminology used in IT and OT cybersecurity domains to enhance situation awareness of CI cybersecurity.

2.2 Reference Architecture for Smart Grid Cybersecurity

Reference models are widely used in system modeling or model-based system engineering to support security-driven analysis (Cloutier et al. 2010). Reference models capture the typical topological structure and functional connections of the architectures. Studies in reference models have been undertaken in the CI field and its sub-field smart grid to provide formal and explicit guidance in the design of critical architectures. Some national or international efforts into smart grid standardization include IEEE P2030 Smart Grid Interoperability Framework (IEEE 2011), EU Mandate M490 SGAM (Gottschalk et al. 2017), etc,.

Some studies attempt to enhance both abstraction and extendability of power-grid reference architecture through ontology or meta-modeling. Irlbeck et al. (2013) propose a bottom-up reference architecture for smart grid, and also discuss the challenges and objectives to create such reference architectures in Europe. Bytschkow et al. (2014) present a CPS reference framework which is then applied in smart grid and automotive domains to model cross-domain dependencies. Korman et al. (2016) provide a reference architecture to combine advanced metering infrastructure and cybersecurity analysis, whereby the reference model acts as an instance of their proposed meta-model. Their smart metering reference model follows UML syntax and allows further implementation using OCL or P2AMF to achieve automated EA (enterprise architecture) analysis. They suggest that a reference model alone cannot meet all the requirements like availability, flexibility, and expressing validation constraints. Instead, reference models together with a modeling tool can meet these requirements at a higher level. They further improve their smart-grid reference model to provide functional and data-flow-oriented reference architecture models to automate security evaluations and cyber-attack simulations. Their work mainly focuses on the cyber network model for smart metering and load balancing related functionalities. The European SEGRID project (SEGRID Consortium 2017) also provides valuable reference models for smart grids, but only focuses on the communication and enterprise modeling, leaving out the physical components. They also provide guidance on the controlling network and related components, but in a rather limited way.

Most of the existing power-grid reference models attempt to provide concrete architecture snapshots. However, the flexibility and compatibility (Cloutier et al. 2010) of the system structure may not be the best designing criteria. Nevertheless, the extensibility of a reference architecture is vital to allow amendments to the model that brings the proposed architecture up to date. Our power-grid cyber-physical reference models support extensible and efficient usage with standardized virtual replicas for cyber connections, cyber-physical setup, and physical processes.

2.3 Dependence Analysis in Cyber-Physical Systems

A complex CI is a system of systems (or SoS) that integrates a collection of devices to achieve desired capabilities (Uslar et al. 2019). In addition, there are complex interaction dependencies between interconnected components (Kong 2019). The dependencies in such a SoS are divided into inter- and intra-dependencies. The inter-dependencies and intra-dependencies of CIs such as smart grids implicitly determine the cascading effects and the system resilience under potential attacks or failures (Marashi et al. 2017).

Akbarzadeh and Katsikas (2021) suggest an application of modeling and simulation methods to study CPSs and detect dependency chains. They also provide an approach to identifying and analyzing inter-dependencies and intra-dependencies between subsystems of a complex system by quantitative measures of the impact of dependency, susceptibility of dependency, and weight of dependency. Besides the study by Akbarzadeh and Katsikas (2021), valuable researches have been carried out for modeling dependencies in CIs in terms of cybersecurity enhancement of such complex systems (Chopade and Bikdash 2011). Ouyang (2014) reviewed six significant types of approaches for modeling interdependencies among CIs, such as empirical approaches and agent-based approaches, and suggested the necessity of an open modeling framework to allow adjustment of CI models. König et al. (2019) propose a combination of local and global views and illustrates the common practical division of the physical and cyber domains. Their work uses a small set of data items, such as assets, interdependencies, and relationships between assets, events, and alarms associated with assets. The physical and cyber parts require these items of the system. They also provide a high-level description of how these parts interoperate, which expands awareness from "knowing what"to "knowing what will happen next", thus solving the core responsibilities of effective risk management. Kwasinski (2020) studies the network and physics of the power grid dependence within the domain and confirms the cyber-physical properties of the power grid. This study shows that internal dependence reduces the resilience of the power system, while service buffers (such as energy storage or data connection re-establishment wait times) help to limit the impact of internal dependencies on resilience. Therefore, the understanding and discovery of internal cyber-physical dependencies are essential to the security analysis

of complex CPSs (Chopade and Bikdash 2011). Actually, the lower resilience in the cyber domain vertices is more critical than lower resilience in physical domain vertices (Kong 2019).

In this paper, we modeled the intra-dependencies within CIs to analyze the interactions between cyber networks and physical controlling networks, while inter-dependencies between connected CIs such as smart grids and water distribution systems are out of scope. Such intra-dependencies are multi-dimensional and are further categorized as functional, logical, spatial, social, and economic dependencies. Functional dependence means that a task of one component is functionally dependent on the other component (Wang et al. 2012; Zhao and Xing 2019). Logical dependence is an implicit correlation between two components, which is commonly seen in software development (Oliva et al. 2011). Spatial dependence describes the propensity that two components with nearby locations have a higher probability of influencing each other. One typical example of spatial dependence is that two physical servers located in the same office have a higher chance of fire propagation. In addition, two software components embedded in the same hardware have dependencies on each other through computing sources competition. Social dependence indicates the impact of social factors such as policies within the energy sector. In contrast, economic dependence is associated with cost or revenues, such as business competition. Social-economic dependence also covers situations where multiple organizations cooperate and are in charge of different sections of the smart grid (Palm 2021). This paper focuses on functional dependencies only, particularly cyber- and cyber-physical functional dependence. Physical dependencies such as power generation, transmission, and distribution are not discussed in this paper, but are included in future works.

We define seven functional dependence rules that are further discussed in Sect. 5. These functional dependence rules further support cascade modeling and criticality analysis. The position of a component in the network system differentiates the importance of the component, and thus contributes to different levels of system failure. Zhu and Milanović (2017) propose a method for weighted modeling CPS introduces a weighted three-dimensional complex network model. The different engineering structures can be modeled without modifications to the topology model in heterogeneous systems. The complex network-based models reveal the vulnerability of different engineering systems and the critical components that could initiate a cascading failure due to the interdependencies between systems. Myhre et al. (2020) apply complex network theory to evaluate the betweenness centrality of the components in a combined electrical grid and ICT systems. They also model the system impact when specific nodes are removed to diagnose important nodes further. The propagation of faults from the network to the physical device will damage the system-level reliability to the greatest extent. Marashi et al. (2017) propose an analytical reliability model that captures the effects of damage from physical and cyber components, as well as the effects of cyber-physical dependencies between these components.

3 Research Method

Our research contributes to the cybersecurity analysis of CIs through a taxonomy that is a model of a domain describing objects that inhabit it. This taxonomy describes an empirically or conceptually derived system of groupings of IT, OT, and physical objects. Thus, it supports the understanding and structuring of the knowledge of CI cybersecurity which is a multidisciplinary area. Next, we introduce our taxonomy and reference model research processes following Peffers et al. (2007) design science research methodology, including problem identification, objectiveness definition, and development, demonstration, evaluation, and communication of the proposed artifact. These activities and corresponding research outputs are summarized in Table. 1.

We first performed a literature review on IT and OT semantics for CI cybersecurity protection, as discussed earlier in Sect. 2. We followed the concept-centric literature review methods proposed by Webster and Watson (2002), and used concept combinations of "critical infrastructure", "cybersecurity ontology", "vulnerability analysis", "power grid", "IT security", "OT security" and "reference architecture" when we searched for studies in *Google Scholar* and *Semantic Scholar*.

We noticed limited taxonomy that is concise and without overlapping concepts and characteristics, partially due to different terminologies used in IT security and OT security. There is also limited support for query-based dependence analysis in previous CI models. And hence, we set up our objective to model IT/OT convergent CI semantics that is extensible in terms of component types and their interconnections, while supporting dependence assessment in a scalable manner.

We define the characteristics of IT and OT entities and their convergence across the cyber-physical layers of CIs, especially smart grids, as CI cybersecurity is a relatively immature domain. Our taxonomy is built upon the semantic models and industrial frameworks introduced in Sect. 2.1, following the Telos (Mylopoulos et al. 1990) language, while also inspired by the architecture analysis and design language (AADL) (Feiler et al. 2003). We define our dependence rules on top of the works introduced in Sect. 2.3. Similarly, instantiations of the proposed taxonomy are

Table 1 Research process following Peffers et al. (2007)

Research activity	Research output		
	Taxonomy	*Analysis method*	Instantiated reference model
Identify problem and motivate	Literature review	Literature review	Literature review
Define objectives of a solution	Literature review	Literature review	Contemporary practices review
Design and development	Literature review	Literature review	Contemporary practices review
Demonstration	Instantiation	Qualitative analysis	Case study
Evaluation	Instantiation	Instantiation	Case study Interview
Communication	To be published in an academic journal	To be published in an academic journal	To be published in an academic journal

built on top of existing smart grid models introduced in Sect. 2.2, which results in multiple reference models to enhance the extendability and explanatory strength (Nickerson et al. 2013). Besides, we follow the Purdue (Williams 1994) enterprise reference architecture model when instantiating objects in the cyber and control layers. The Purdue Model is adopted in internal standards such as *IEC/TS 62443-1-1:2009(E)*. Even though the Purdue Model is used more in manufacturing architectures (Boyes et al. 2018), its structure still applies to the similar cyber and cyber-physical layers in smart grid. We also incorporated our knowledge about the smart-grid architecture, especially the physical layer and its related components and processes, that are gained through our organized workshops, newspaper articles, and discussions with a power-grid company located in Sweden.

In doing so, we derive a diverse set of characteristics and dimensions of CI objects in terms of cybersecurity. We visualize our taxonomy through an open-source tool called ConceptBase (Jarke et al. 1995). This tool supports the representation of classes, domain-specific objects, and instantiated models in the same database. It also allows the specification of graphical symbols for specific classes, which is then applicable to all instances of those classes. The defined models can also be easily extracted in a preferable format (e.g., XML format) and employed in power-grid simulation and vulnerability modeling. We validate the structural, functional adequacy, compatibility, and coverage of our taxonomy in two case studies (Yin 2009) of instantiated power grid models, following standards of Nickerson et al. (2013) as well as the ontology quality evaluation and requirements (OQuaRE) (Duque-Ramos et al. 2014) framework. The OQuaRE framework adopts the ISO/IEC standards for software product quality requirements and evaluation (Suryn et al. 2003) in ontology assessment. We also conducted a series of semi-structured interviews (documented by video recording and anonymized for privacy concerns) to evaluate the usefulness of our artifacts from the perspective of application users.

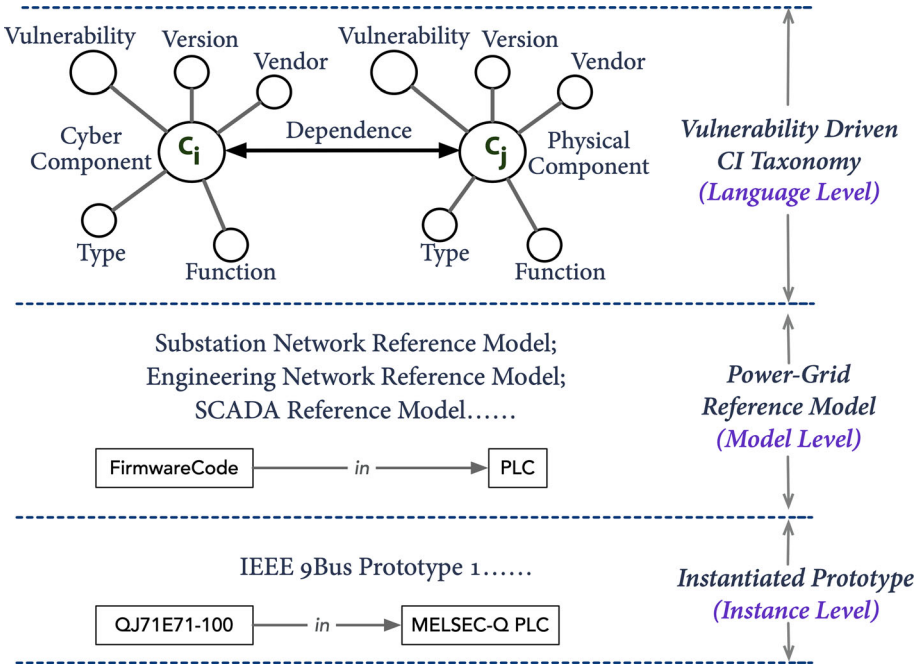
4 Artifact I: Taxonomy for Critical Infrastructure Cybersecurity Analysis

Before digging into the details of our taxonomy and instantiated models, we define the interactions between our proposed taxonomy and instantiated models, as illustrated in Fig. 2. We classify different levels of our models based on meta-modeling layers discussed by Jeusfeld et al. (2009). A meta-model consists of formal statements that clarify semantically related classes about the models. Our vulnerability-driven taxonomy defines high-level classes like *Component*, as well as shared class-level methods, attributes, and constraints. In CIs, component types making up IT, OT, or cyber-physical system fabric can be identified as concepts in the taxonomy. For example, this taxonomy classifies CPS elements into semantic modules that are used to compose a CPS model as a network of cyber and physical components.

Based on the taxonomy, we established smart grid models following generic constructs, constraints and rules of each domain. These established models summarize common structures of power-grid systems to allow a higher level of reproducibility. We then further use these models to create prototypes of power-grid that follow the same constructs, instead of modeling a power-grid system from scratch.

4.1 Cyber-Physical System Semantics

Figure 3 illustrates the top-level structure of our taxonomy that starts with “ComplexObject”. “ComplexObject” is the most general class, and can subsume any object. For



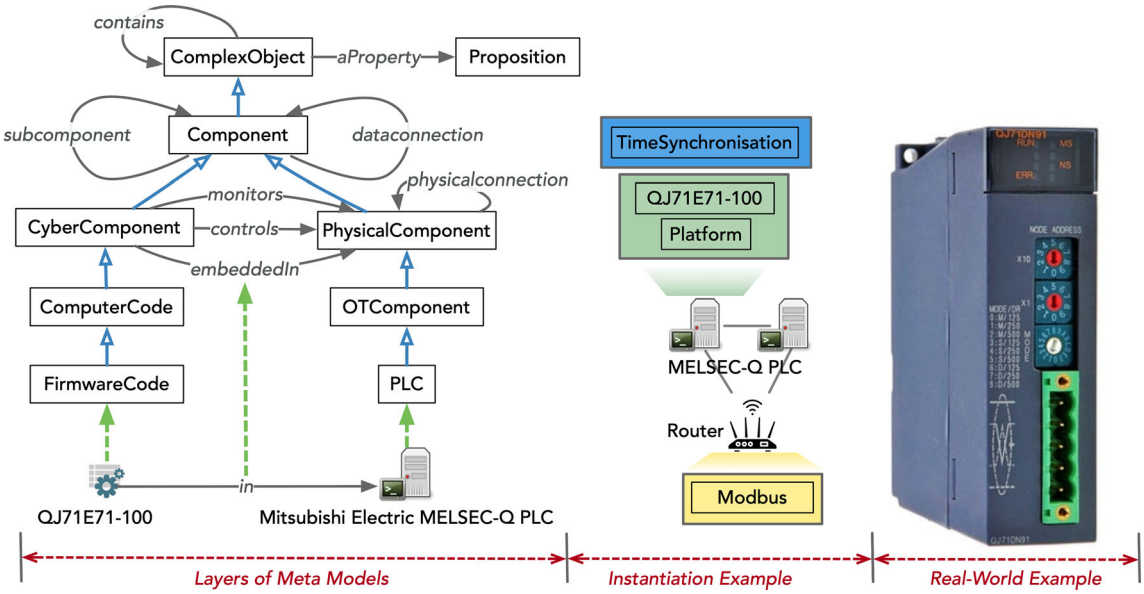


Fig. 3 Top level taxonomy and instantiation example. (Subclass relations are denoted by blue arrows with white arrow heads. Instantiating relations are visualized by green broken links.)

instance, we define a sub-class "Component". "ComplexObject" has a relation "property" to "Proposition", which is used to attach various properties to power-grid components. A component has two relations, namely data connections and sub-component configuration. Data connections refer to specific flows like data flows or control flows that bridge two given components. Besides, a component is decomposed into sub-components. For example, a RTU (remote terminal unit) device is decomposed into hardware and the embedded firmware.

Components subsume physical, cyber, or network components, as presented in Fig. 4. Cyber components are embedded in physical components which have certain geographical locations. Therefore, physical components and embedded cyber components share the same physical connections. A network component is defined as an organization where a certain set of components follow a common set of rules for access and management. Our taxonomy consists of facts, constraints, types, and security-related attributes. For example, we define each

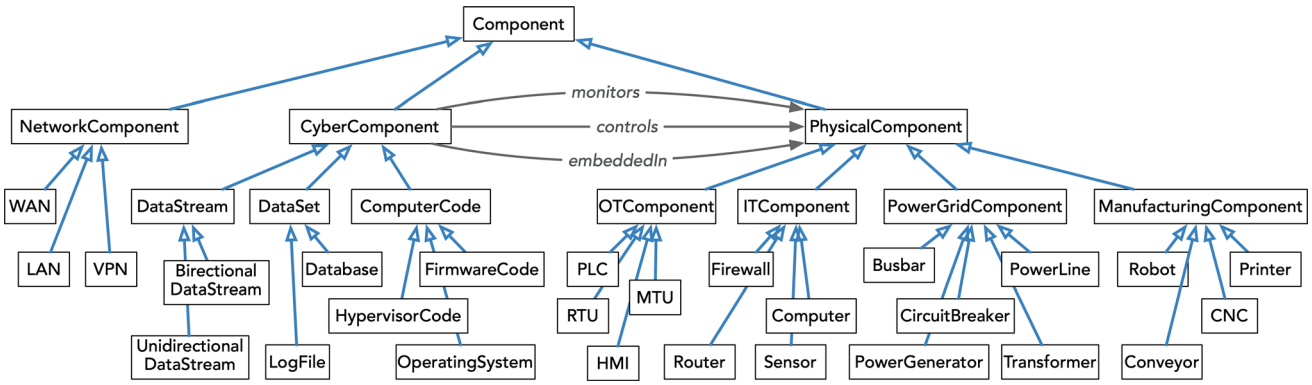


Fig. 4 Cyber, physical and network components in the cyber-physical taxonomy

component's properties to clarify its vendor, product model, version, build number, and protocol. We may instantiate a Mitsubishi Electric (2022), provided by the vendor *Mitsubishi Electric*, with adopted protocol *Modbus*.

The physical component is also extended in our taxonomy to represent a range of IT and OT components used by SCADA and SIEM (security information and event management) services at the cyber and control layers. IT, OT, and power-grid components are distinguished by the cyber components embedded in these physical components. RTU, MTU (master terminal unit), HMI (human–machine interface), and sensors support the SCADA control system, along with routers and optical networks (Humayed et al. 2017). We categorize these components as OT components. Firewalls and endpoint security tools support SIEM's data analysis and correlation, both of which belong to IT components (Vielberth et al. 2020). More specifically, MTU periodically initiates and acquires RTU data and allows operators to perform control tasks remotely. RTU directly collects field information like process data and variables from sensors and deploys commands through actuators. HMI can be either standalone terminals or embedded in other devices like MTUs. Meanwhile, RTU and MTU are connected to other SCADA components like SCADA servers through routers, optic cables, and switches (Boyer 2009; Stouffer et al. 2011).

Cyber components subsume computer code and data sets captured at the cyber-layer level. For example, SCADA programs are embedded into micro-controllers to monitor some physical power-grid processes (Boyer 2009). Computer code components represent the actual code running and embedded in physical components. Computer code components further subsume firmware code, operating system code, hypervisor code, etc. Firmware code usually runs on the bare metal of the chip and supports the low-level control of the hardware. One example is HMI firmware which contains graphical libraries where graphical symbols with tag names are associated with specific devices and parameters of the devices, such as a particular switch and the ON/OFF status of the switch. An operating system controls the central host computer hardware and facilitates interactions between hardware and software components. Hypervisor code virtualizes the hardware that runs kernel-model processes. We can also specify the configuration between an operating system and a hypervisor as bare-metal or hosted hypervisors.

A communication or corporate network comprises relay stations like routers, switches, firewalls, and endpoints like computing servers. Routers and switches usually have access to most network segments and have prime positions for data exfiltration. Switches parse and handle many Layer 2 protocols that are normally enabled by default on all of the available ports of the switches. A network component follows a specific protocol that is a set of rules, syntax, and semantics that allow data transmission between two or more entities. Network components further subsume WAN (wide area network), LAN (local area network), and VPN (virtual private network).

Cyber components further subsume data stream components. Moreover, the data stream subsumes bidirectional data stream and unidirectional data stream. Data stream is a critical concept in our taxonomy through its contribution to the system dependencies. A data stream object involves at least two components as participants, and requires one of the participants to be the initiator. Usually, two participants communicate through a master–slave mechanism, namely, a master device that initiates queries, and a slave device that responds with requested data to complete transactions. One participant acts as the sender for a unidirectional data stream, while the other participant acts as the receiver. In comparison, a bidirectional data-stream sender functions as the receiver in a reversed direction. The data stream definitions are specially useful for SCADA automation analysis (PES 2008).

Figure 5 shows two instances of data streams. The example at the top is a unidirectional data stream. A time-

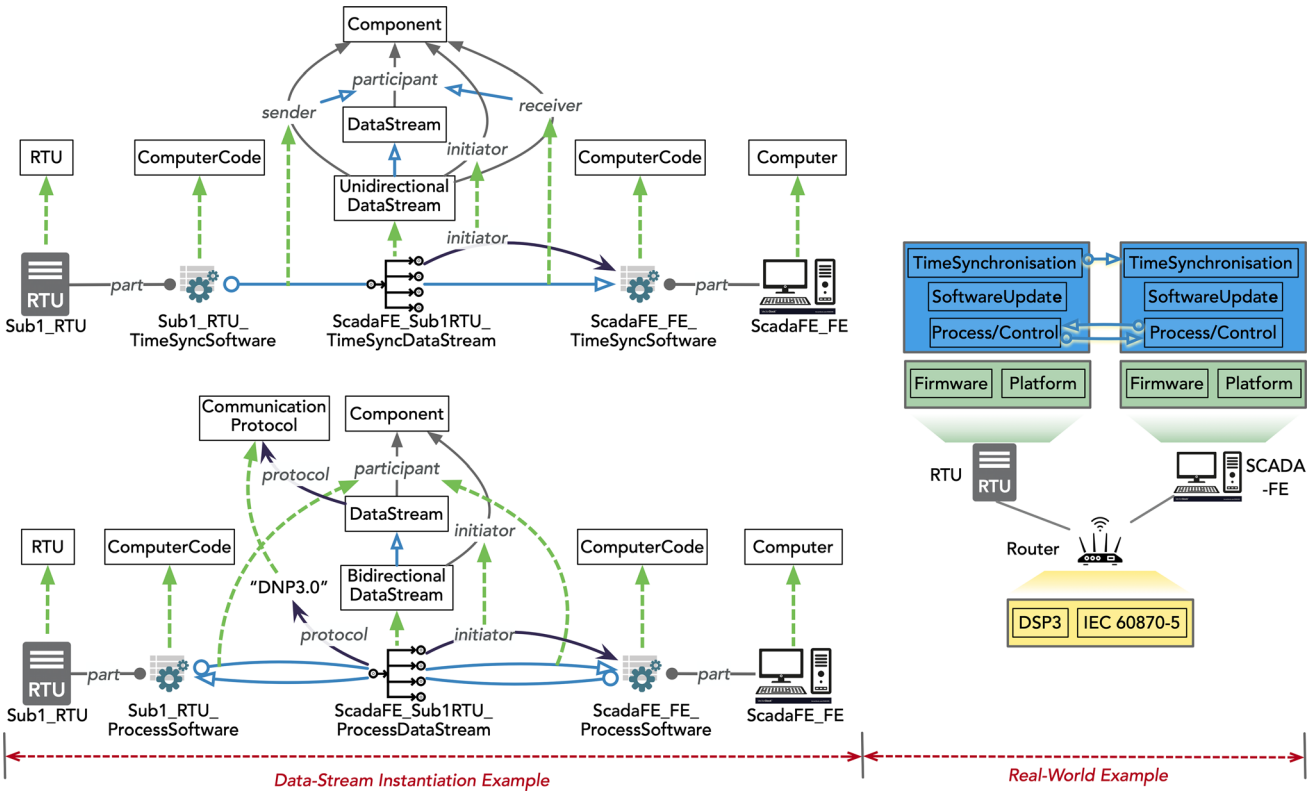


Fig. 5 Unidirectional and bidirectional data stream example

unit software embedded in the SCADA front-end (FE) server initiates a time synchronization request to a client time-synchronization software embedded in a RTU in the Sub1 network. This client RTU software checks the request and sends back the data. The other example at the bottom illustrates a bidirectional data stream for process and control commands. SCADA FE receives the power process data from Sub1 RTU and further passes it to SCADA for monitoring and analysis, returning control commands. Then, SCADA FE transmits the control commands back to Sub1 RTU and then delivers them to actuators.

4.2 Vulnerability-Driven Cybersecurity Semantics

We further define *Vulnerability* that is subsumed under *ComplexObject*. *Vulnerability* exists in *Component* that matches the product configuration affected by this *Vulnerability*. *Vulnerability* further has attributes, including metadata, tracking, weakness information, severity, threat, related attack, and corresponding remediation. These attributes are clarified by extending the schemas provided by CVRF that are commonly used by repositories like CVE and vendors like *Cisco* and *Microsoft* to support security information standardization and sharing.

More specifically, the *Metadata* attribute wraps up basic information like *CVE-ID*, *CVE* report description, references, and reports provided by vendors and other security analysts. *Tracking* attribute stores time-related information like publication dates in various data sources, reflecting the vulnerability in the lifecycle. *Affected Product* attribute takes in security-related software flaws, misconfigurations, and other vulnerable configuration information. *Threat* attribute gathers threat types that the vulnerability may be exploited, which are one or more categorical threat types in *cvedetails.com*. *Weakness* attribute collects information concerning weakness patterns such as MITRE (2022d) (*CWE*) terminology in the investigated vulnerability. Similarly, *Attack* attribute aligns the vulnerability to the attack patterns such as MITRE (2022a) (*CAPEC*) identifiers and related tactics, techniques and possible implementation procedures provided by *ATT &CK*.[1] *Severity* class captures vulnerability severity scores and matching vectors under the FIRST (2022) (*CVSS*) V2 and V3 mechanisms. And lastly, *Remediation* class provides mitigation suggestions provided by vendors and third-party security analysts.

Figure 6 illustrates these attribute of *Vulnerability* as well as an instantiated example with *CVE-ID* as MITRE (2021). This vulnerability instance indicates a weakness of improper authentication and a *CWE* entry as *CWE-287* that

[1] https://attack.mitre.org/.

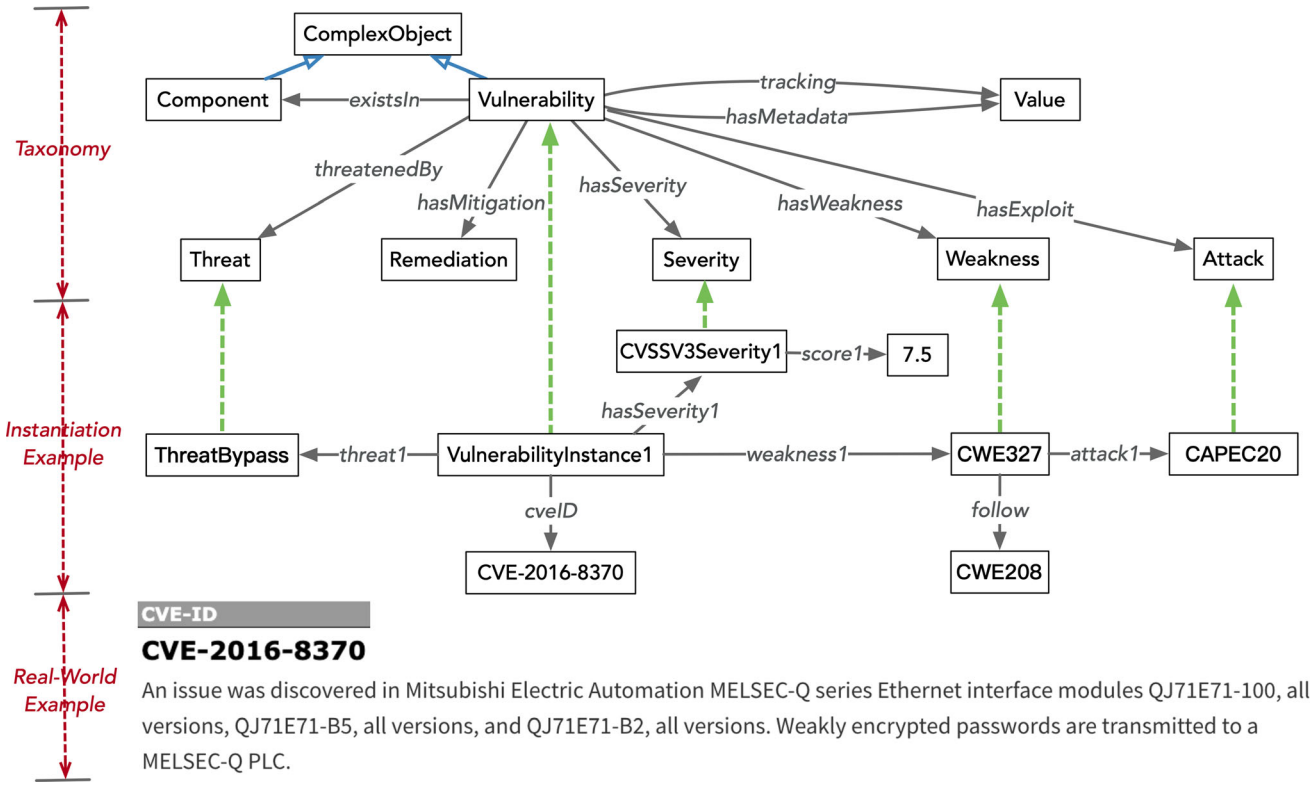


Fig. 6 Taxonomy of security objects

further follows insufficient session expiration with entry *CWE-613*, and may be exploited by an authentication bypass attack with a *CAPEC* entry *CAPEC-114*.

5 Artifact II: Model-Based Dependence Analysis and Vulnerability Assessment Method

We further propose a model-based security engineering method that has our taxonomy in the central role, to support dependence analysis and vulnerability assessment, as illustrated in Fig. 7. More specifically, CI models instantiated through our taxonomy deliver structured system configuration information to support vulnerability queries. Subsequently, vulnerability analysis methods are used to identify severity, threat, and weakness labels to these CI vulnerability instances. Then, these vulnerability instances with added security indicators are inserted into the established CI models to support further visualization and static query-based analysis.

5.1 Cyber and Cyber-Physical Functional Dependence

Here we define functional dependencies (or FD) as: If component C_i depends on component C_j to complete its functional activities properly, then we say that component C_i has functional dependence $FD_{(i,j)}$ on component C_j. We further define seven FD rules as depicted below, which are employed to describe the complexity of a software component. These seven rules can be used to define system dependencies using the static system configuration information.

1. FD Vertical Rule V^1: If a cyber component C_i is embedded in an IT or OT component C_j, then C_i is functionally dependent on C_j, or $FDV^1_{(i,j)}$.
2. FD Vertical Rule V^2: If hypervisor or operating system component C_i contains cyber component C_j, then C_j is functionally dependent on C_i, or $FDV^2_{(j,i)}$.
3. FD Horizontal Rule H^1: If an OT component C_i contains a cyber component C_k that collects process data from a physical component C_j, then C_k is functionally dependent on C_j, or $FDH^1_{(k,j)}$.
4. FD Horizontal Rule H^2: There exists control data from a cyber component C_i (embedded in an OT component C_k) to a physical component C_j, then C_j is functionally dependent on C_i, or $FDH^2_{(j,i)}$.
5. FD Data Rule D^1: There exists data stream between two cyber components C_i and C_j, and C_i is the receiver of the data stream, then C_i is functionally dependent on C_j, or $FDD^1_{(i,j)}$.

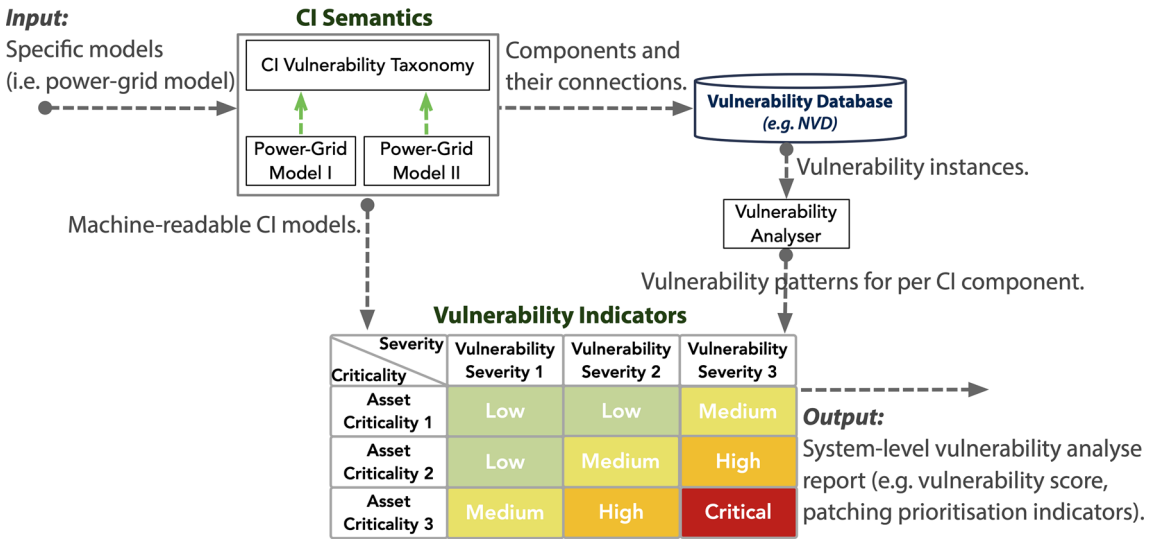


Fig. 7 Bird view of vulnerability assessment method

6. FD Data Rule D^2: There exists data stream that listens to dataset C_j, and C_i is the receiver of the data stream, then C_i is functionally dependent on C_j, or $FDD^2_{(i,j)}$.
7. FD Network Rule N^1: If a server computer C_i is connected to a network through a router component (or a switch component) C_j, then C_i is functionally dependent on C_j, or $FDN^1_{(i,j)}$.

All rules are implemented as deductive rules in the ConceptBase system. Following these dependence rules, we conduct some static analysis on *SCADA* and *Substation* based on the aforementioned reference models. The red dashed lines highlight the functional dependence. Partial SCADA network contains *SCADA_Historian* and *SCADA_Server*, *SCADA_FE* workstation and a *RTU* in one substation, as illustrated in Fig. 8.

In the physical server, hardware components integrate with operating system software and manage PC storage. Meanwhile, a hypervisor host deploys and serves virtual systems, which provides an abstraction layer for virtualization. For example, the hypervisor in *SCADA_Server* provides virtual machines for three system packages, namely *SQL*, *Control* and *Office* systems. The *SQL* system contains a database engine that processes queries and manages database files. The *Control* system stores and retrieves power-grid process and control data using queries. The *Office* system provides maintenance and service to the power-grid software, firmware, and configurations. According to the vertical FD rules, the application (APP)

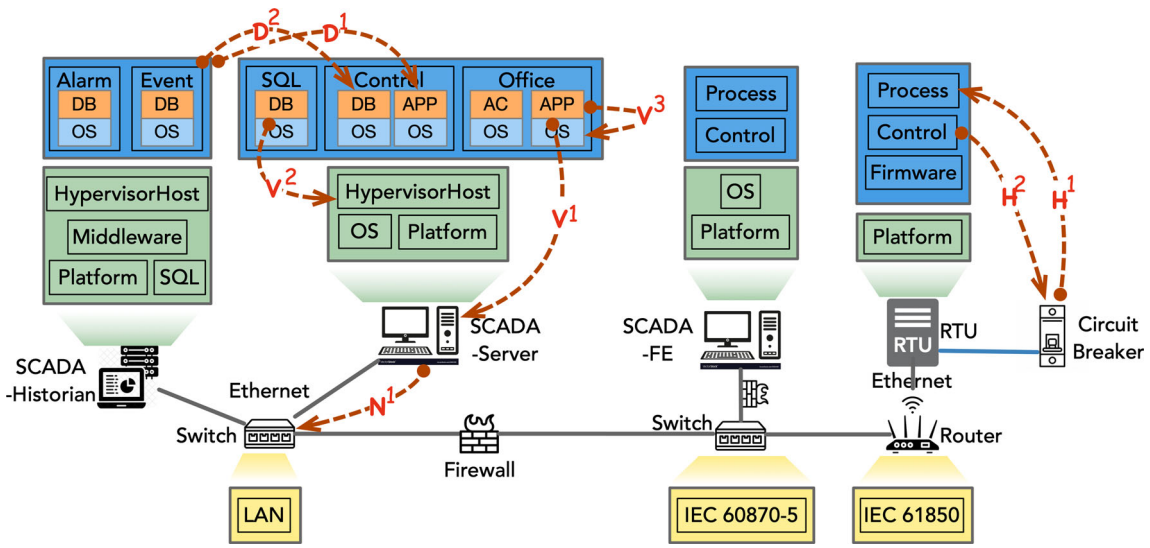


Fig. 8 Dependence-analysis example of Sub1 RTU

server in the *Office* system is vertically dependent on the guest operating system (marked as V^3) and the physical server (marked as V^1). The database (DB) server in the *SQL* system is vertically dependent on the hypervisor host (marked as V^2). We also show the dependencies due to process and control data between *RTU* and *CircuitBreaker*, which are marked as H^1 and H^2.

Figure 8 highlights the historical analysis data stream example marked as D^1 and D^2. D^1 illustrates the dependence of the data stream receiver, the *Event* system in *SCADA_Historian*, towards the data stream sender, the application server in *Control* system of *SCADA_Server*. D^2 presents the dependence of the same receiver on the listened database.

On top of defined correlations between component nodes, we further define dependency matrix $FD_{(i,j)}$ between C_i and C_j. Such a dependency matrix supports analyzing nodes' centrality and influence levels. Meanwhile, dependence rules assist CPS cascade modeling, which is introduced next.

5.2 Cascading Modeling and Criticality Analysis

Cascading is a propagation behavior demonstrated by a chain of events/failures in a system. Failure refers to the state or condition of not meeting a desirable or intended objective, and can be generated by external factors like attack or failure from neighbor components. Failure can happen on any or multiple components. Cascading failure starts somewhere in the system, which in turn causes a new failure in a different component (Guo et al. 2017; Vaiman et al. 2012). We further define the following rule to support cascade modeling, which are extended to the transitive closure: "There exists a failure or compromise of a component C_i that a component C_j is functionally dependent on, then the failure would probably propagate to C_j ".

Here we claim that the failure propagation from C_i to C_j has a certain probability, considering that system configurations or network structures with proper security compliance reduce such probability. In the case studies of this paper, we assume that such probability is equal within the system and leave weighted probability analysis as future works.

We calculate the number of components that have direct functional dependence on component C_i, and define it as $N_{i,j}^{FD}$ where $0<i,j<M$ (M is the number of components). Such a component is a critical function point with higher criticality.

5.3 Vulnerability Retrieval and Feature Allocation

We retrieve vulnerability instances for specific CI components using their name, version, and vendor information. Besides, build numbers (like the build numbers for Windows server), release numbers, and cumulative security update KB package numbers (like the KB numbers for VMware products) are extracted to track the system update history. The abovementioned data is integrated into our model, and is necessary to generate snapshots of system configuration information that are later matched against online vulnerability databases like CVE and NVD. The retrieved vulnerability instances from these repositories contain *CVE-IDs* that can be used to extract further the corresponding weakness, threat, and attack labels. The process of identification and retrieval of vulnerability features are illustrated in Fig. 9, while using vulnerability instance *CVE-2021-36745* as an example.

More specifically, *CVE* and *NVD* reports contain references to the affected vendors and third-party analysts. These references contain URLs that can be fetched to scrap information from vendors' and security analysts' websites. With the vulnerability *CVE-IDs*, URL links for additional third-party analysts are also accessible. Figure 9 presents an example that uses *CVE*-IDs to crawl the specific link within the *cvedetails.com* domain, and then scrap vulnerability reports to fetch threat category information. Simultaneously, *CWE-IDs* are fetched from the vulnerability reports and are used as tags to retrieve the *CWE* version 4.6 document for the matching attributes for these *CWE-IDs*, particularly names, descriptions and correlated *CAPEC-IDs*. These fetched *CAPEC-IDs* are further used as tags to query the *CAPEC* version 3.6 dataset for the corresponding names, descriptions and *ATT &CK-IDs*. Similarly, *ATT &CK* names and descriptions are extracted from *ATT &CK* version 10 document with the list of retrieved *ATT &CK-IDs*, as shown in Fig. 9. It is possible that different *CWE-IDs* are assigned by *NVD*, vendors, and other security analysts. And hence, labels are added to the features to differentiate the feature sources.

6 Instantiating the Taxonomy in Power-Grid Reference Modeling

Our taxonomy can be extended to enhance expressiveness in a specific domain. Figure 4 illustrates some partial examples of cyber and physical components in our power-grid taxonomy. Considering the functionalities of power-grid systems, we define that physical components further subsume power-grid components that are deployed for electric power generation, transmission, transformation, and distribution. A physical component has spatial

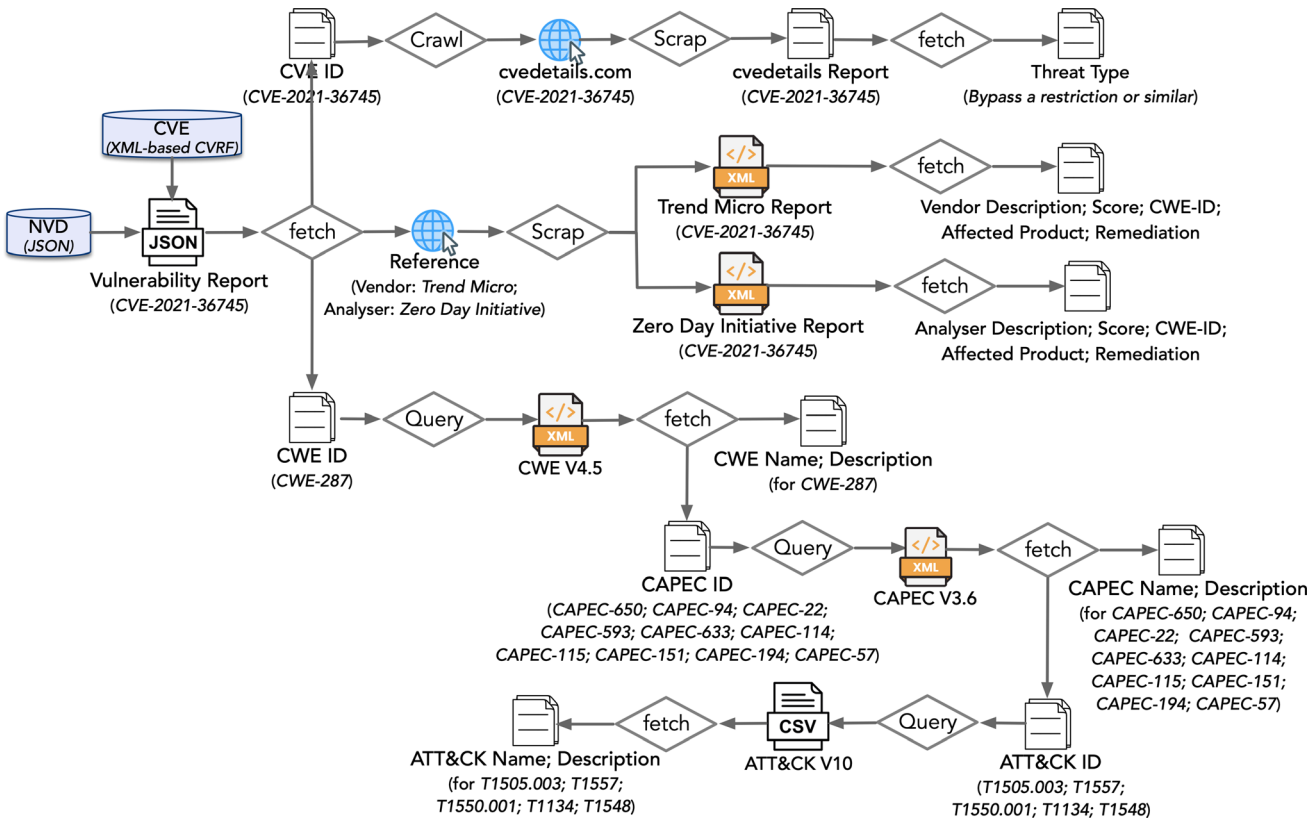


Fig. 9 Example of data correlation for vulnerability *CVE-2021-36745*

geographic property or a specific location. Meanwhile, power-grid components have power connections. An example of power-grid components is a circuit breaker employed to disconnect a power transmission. Another example is a transformer that transfers electric power between two electric circuits. We also define functional requirements for power-grid components like voltage and connected power lines.

Figure 10 shows a power substation as an instance of our taxonomy that takes references from Knapp and Samani (2013). Only part of the instances is shown to ensure readability. A power generator *Generator1* has power connection with *Busbar01* which is connected to the second busbar *Busbar02* through a transformer *Transformer1*. *Busbar02* also has power connection with power line *PowerLine02_03*. On top of *Substation1*, an operation network *Sub1* covers the power process control and monitoring. IT components (i.e., *Sub1_RTU* and *Sub1_Workstation*) are connected to the power-grid components (i.e., *Busbar01* and *Busbar02*) through data connections (i.e., fibre). *Sub1_RTU* and *Sub1_Workstation* are also connected through a data stream for local maintenance. This unidirectional data stream has access to the maintenance data store embedded in *Sub1_RTU*.

We model three layers of networks to identify interdependencies across CIs, namely the cyber, control, and physical layers. Each layer incorporates different functional sections or zones of CI networks. Network zones are connected through routers and are protected by firewalls.

6.1 Public Internet and Other Networks

The cyber layer includes a general internet area and the power-grid enterprise network. The wide internet area contains a *CustomerService* network, an *Analyzer* network, a *Vendor* network and an *EnergySupplier* network, as illustrated in Fig. 11.

The *CustomerService* network contains a server computer embedded with two software packages, namely *CustomerManage* software that regulates customers' power consumption, and *Analyzer* software that works for load prediction (Abubakar et al. 2017). These two packages can also be integrated into one module, such as the ABB (2022). Multiple vendors provide different software and hardware that meet diverse access, operational and technical requirements of the smart grid. Some of these vendors require privileged remote network access or *VPN* (Virtual Private Network) tunnels to support, maintain or troubleshoot certain technologies and systems inside the smart

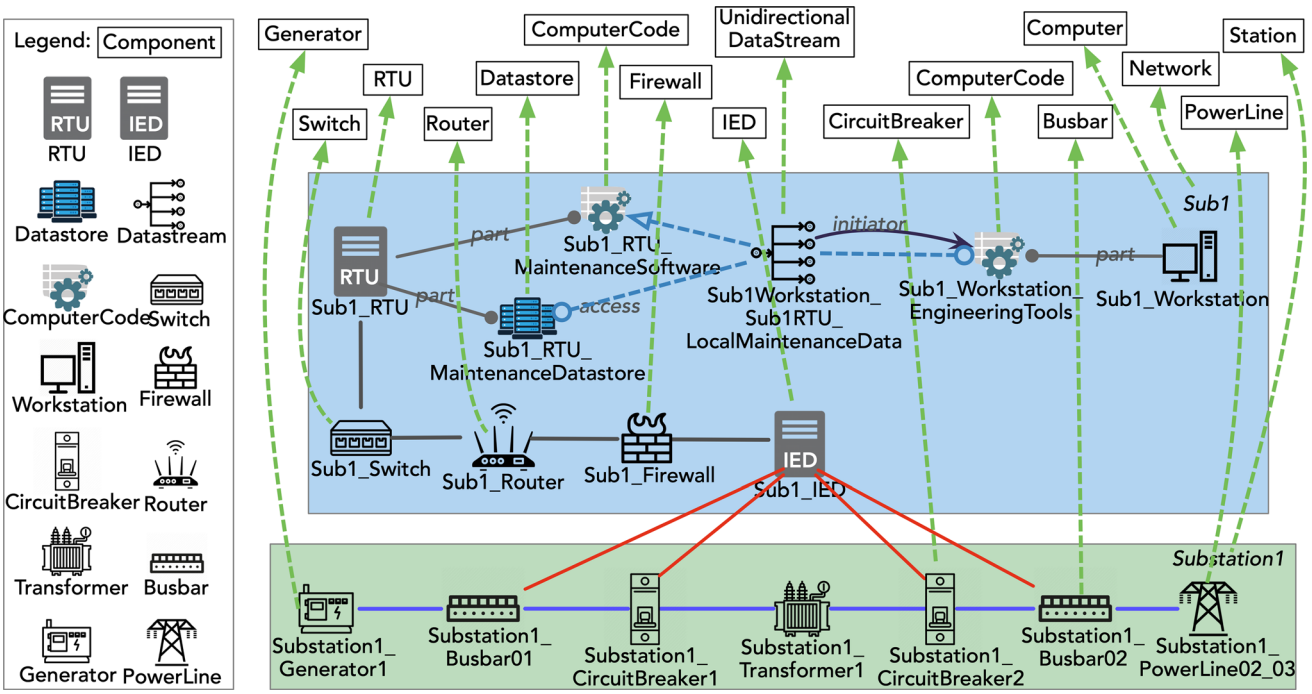


Fig. 10 Instantiated power-grid substation example. (Green dashed arrows represent instantiating. Grey solid lines and purple solid lines represent data connections and power connections, separately. Grey lines marked with "part" represent system configurations, meaning one component is subsuming the other component.)

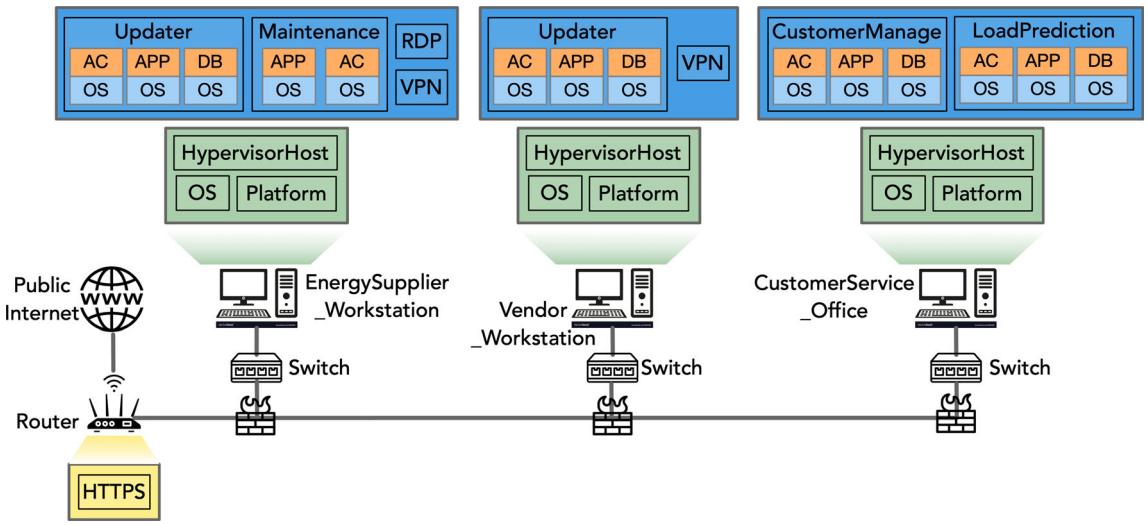


Fig. 11 Reference model of public Internet and other networks

grid (Zeinali and Thompson 2021). *EnergySupplier* network refers to external power suppliers' business administration and marketing management. Sometimes energy suppliers employ VPN or RDP (remote desktop protocol) access to remotely log into DER (distributed energy resource) substations for monitoring and operating purposes (Ying et al. 2014). *Updater* refers to the software package, IO (i.e., input and output), cumulative update file, and other related programs needed to manage the updating of hardware, software and firmware components. *Maintenance* covers the necessary programs used to handle system configurations.

Normally, the *Vendor* network, the *CustomerService* network and the *EnergySupplier* network belong to different stakeholders. These stakeholders get access to the power-grid enterprise network through an intermediary, i.e., the *PublicInternet* network, which supports internet applications like web browsing.

6.2 Office, Engineering, and Security Operating Center Network

The enterprise network contains an *ITAdministration* network for IT administration management, an *Engineering* network for system maintenance and configuration update, a *SOC* (security operation center) network for security-related analysis and safety inspection, as well as an *Office* network for local office operation, as illustrated in Fig. 12.

The *ITAdministration* network is connected to public internet servers via firewalls. *ITAdministration* network is further connected to *Engineering* network, *SOC* network and *Office* network through router and firewalls. The *ITAdmin*, or IT administration network, is in charge of network operating and also preventing and fixing network problems locally or through RDP. Besides, mail administration and network administration are utilized to maintain and configure network and mail routing, separately. The *SOC* network involves system monitoring and risk management, as well as control and digital forensics. A typical tool used in SOC is SIEM, which leverages advanced analytics for incident response and SOC automation (Vielberth et al. 2020). *Office* network contains a local office server that oversees mail configuration, remote desktop software, and web browser. *Engineering* network contains a local server that undertakes business-driven investigation and SCADA statistic analysis. *Engineering* network also covers a workstation that supervises system software updating and maintenance.

6.3 Control Center Network

The control layer includes two networks, namely a control center and a *SCADA WAN* (wide area network). *Control Center* mainly involves *SCADA* for process data monitoring, control command distribution, and power process synchronization, as illustrated in Fig. 13. Various intelligent grid devices are connected to the control center from power generation substations, high and low voltage transformation substations, distribution assets, and the distributed controlling workstations. The *SCADA_Server* monitors and controls these distributed substations (Knapp and Samani 2013). The real-time power process data is virtually presented on *SCADA_HMI* and then further transmitted from *SCADA_Server* to *SCADA_Historian* for statistical analysis. Furthermore, system update and maintenance data is transferred to and stored in *SCADA_FTP* before direct usage in the controlling servers. *SCADA_Timer* is in charge of the time synchronization of the whole system (Boyer 2009; Stouffer et al. 2011).

SCADA WAN is a shared network between *SCADA-FrontEnd (FE)* server and distributed substation networks. *SCADA-FE* server manages event-based communication with the field devices, and is therefore responsible for processing and controlling data transfer. Namely, *SCADA-FE* works similarly as a master station and requests data periodically from field devices like RTUs.

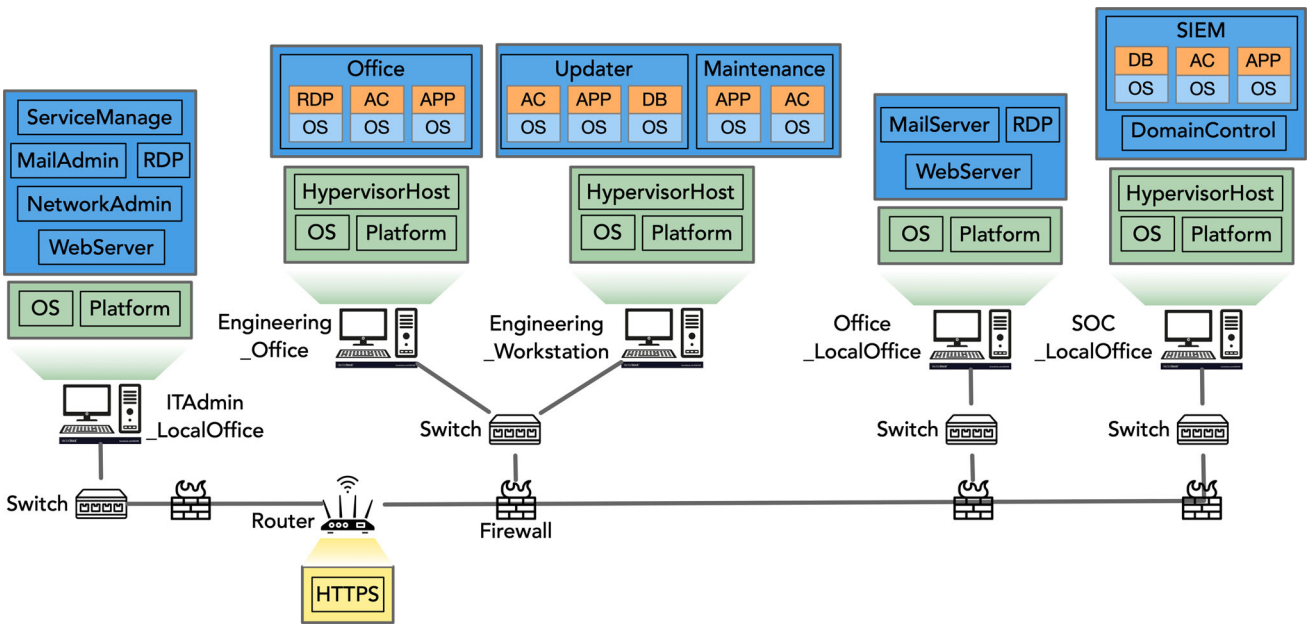


Fig. 12 Reference model of office, engineering and security operating center network

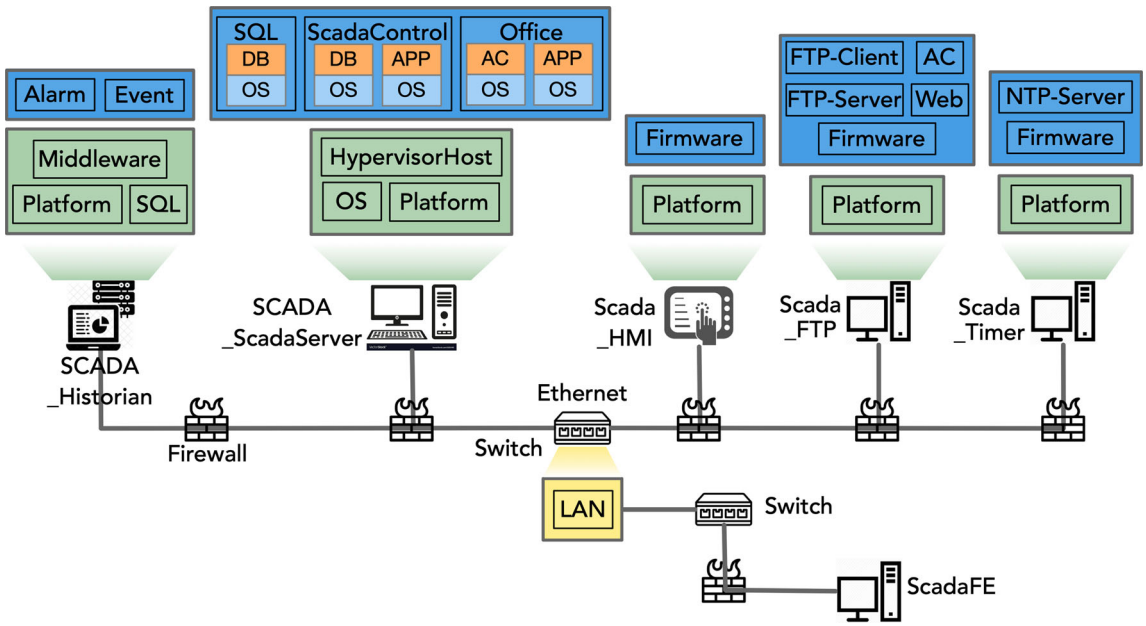
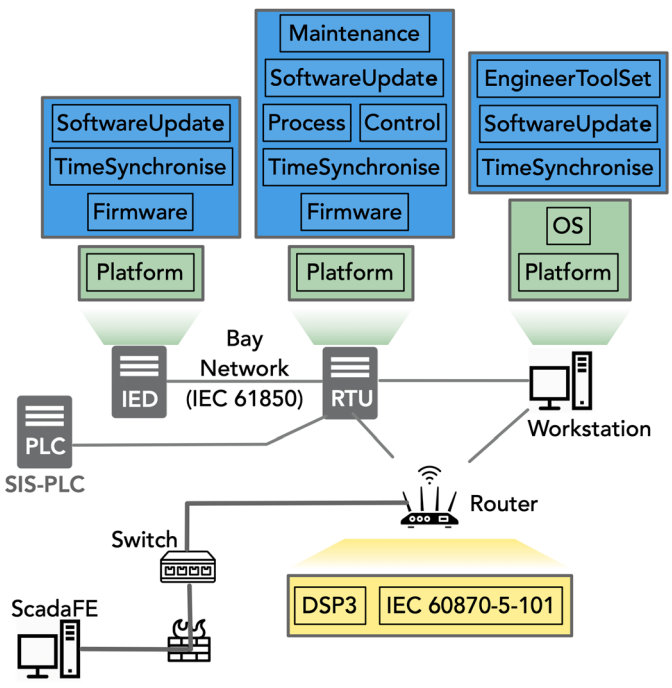


Fig. 13 Reference model of control center network

6.4 Substation Network

Substation network includes LANs between RTUs and local workstations, as well as *Bay network* that lies in the interface of the control layer and physical layer, as illustrated in Fig. 14. *Bay* control IEDs provide flexible control and backup protection for physical components such as circuit-breakers and earthing switches (Brand et al. 2003), which normally follows the *IEC 61850* communication protocol (Brand et al. 2011). In the physical layer, *IEDs* (intelligent electronic devices) and *SIS-PLCs* (safety instrumented system PLC that can enable emergency shutdown) are connected to RTUs. The data connections build the bridge between control units and physical units, based on which RTUs in local substations remotely control and monitor power processes. Such control and monitoring



Partial Instantiation Example:

Sub1 in Network with subcomponent
sub1: Sub1_RTU;
sub2: Sub1_IED;
sub3: Sub1_Workstation
end
Sub1_RTU in RTU with subcomponent
sub1: Sub1_RTU_Firmware;
sub2: Sub1_RTU_MaintenanceSoftware;
sub3: Sub1_RTU_SoftwareUpdater;
sub4: Sub1_RTU_ProcessSoftware;
sub5: Sub1_RTU_TimeSyncSoftware
end
Sub1_IED in IED with subcomponent
sub1: Sub1_IED_Firmware;
sub2: Sub1_IED_SoftwareUpdater;
sub3: Sub1_IED_TimeSyncSoftware
end
Sub1_Workstation in ServerComputer with subcomponent
sub1: Sub1_Workstation_OS;
sub2: Sub1_Workstation_SoftwareUpdater;
sub3: Sub1_Workstation_EngineerToolSet;
sub4: Sub1_Workstation_TimeSyncSoftware
end

Fig. 14 Reference model of substation network

functionalities include devices switch, set-points for generators, and sequential control.

6.5 Power-Grid Substation

The transmission substation is connected to a power generator. This substation is composed of six circuit breakers, two transformers that convert between two transmission voltages, and multiple transmission lines, busbars, and switches. More specifically, two high-voltage switches allow the neural line *NLine1* to be isolated and connected to a grounding system such as a ground fault neutralizer, as illustrated in Fig. 15a.

The transmission and distribution substations have similar structures (Ruland et al. 2017), namely two or more transmission lines as power input, feeders as power output, and one or two transformers in the middle, as illustrated in Fig. 15b. Meanwhile, smart meters are deployed to record electric energy consumption, voltage levels, and other physical process data (Korman et al. 2016). Besides, a communication network provides supervisory process management for this electrical grid. Such a communication network is divided into several distributed LANs connected to the power substations separately, as well as the control center network.

6.6 Data Asset Identification

Identification of information assets is a vital step in the risk management process highlighted in ISO 27000 series (Disterer 2013). The power system produces data that can be turned into valuable information when appropriately processed and encrypted. This information is a valuable asset that benefits optimized investments, accurate problem analysis, and safe utilization of the power system. This paper focuses on the critical data assets of a power-grid system and the containers where the assets are stored, transported, and processed.

6.6.1 Process Data and Process Control Data

Process data refers to the measurements of the power processes collected by distributed sensors. Figure 16 presents the process data periodically polled to the centralized system platform SCADA through the process liaison, SCADA FE. Simultaneously, SCADA application servers such as the Analyzer server compute process data from real-time and historical databases, to generate commands for the SCADA SystemServer. These commands are then sent to distributed actuators to supervise optimal power flow. Meanwhile, process data is transmitted to the safety-inspection server for inspectional analysis like voltage stability assessment. Once unstable power status is

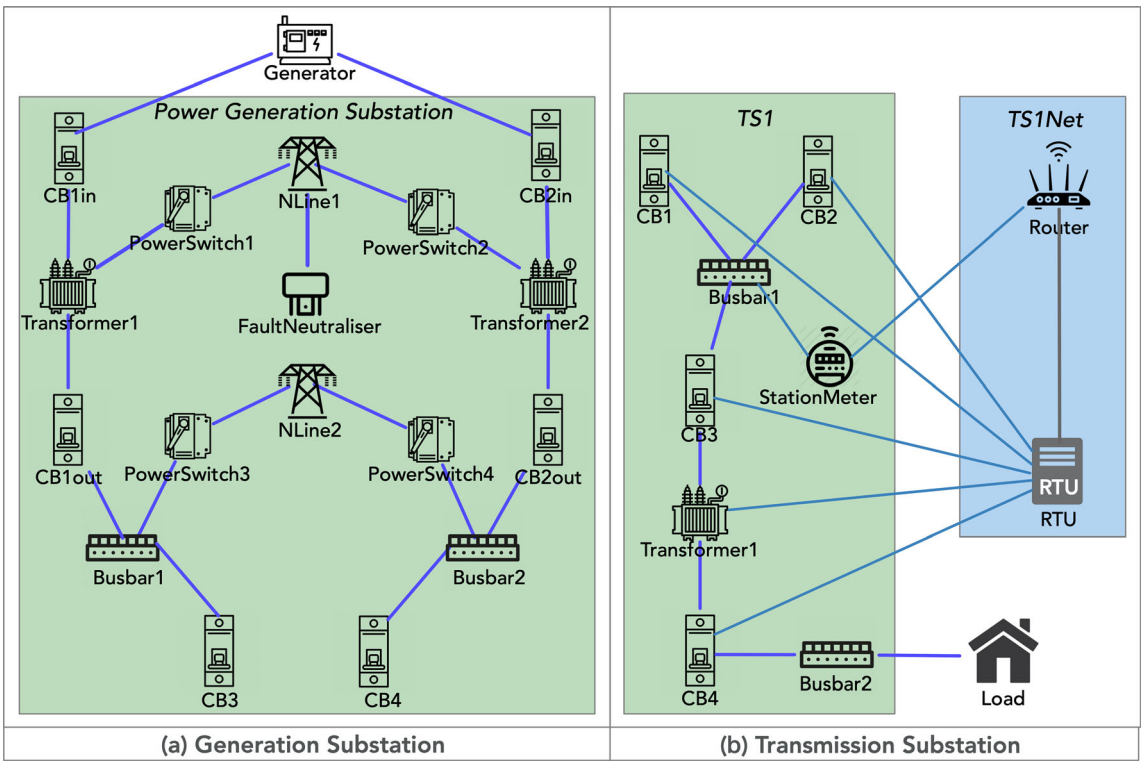


Fig. 15 Reference model of power generation/distribution substation network

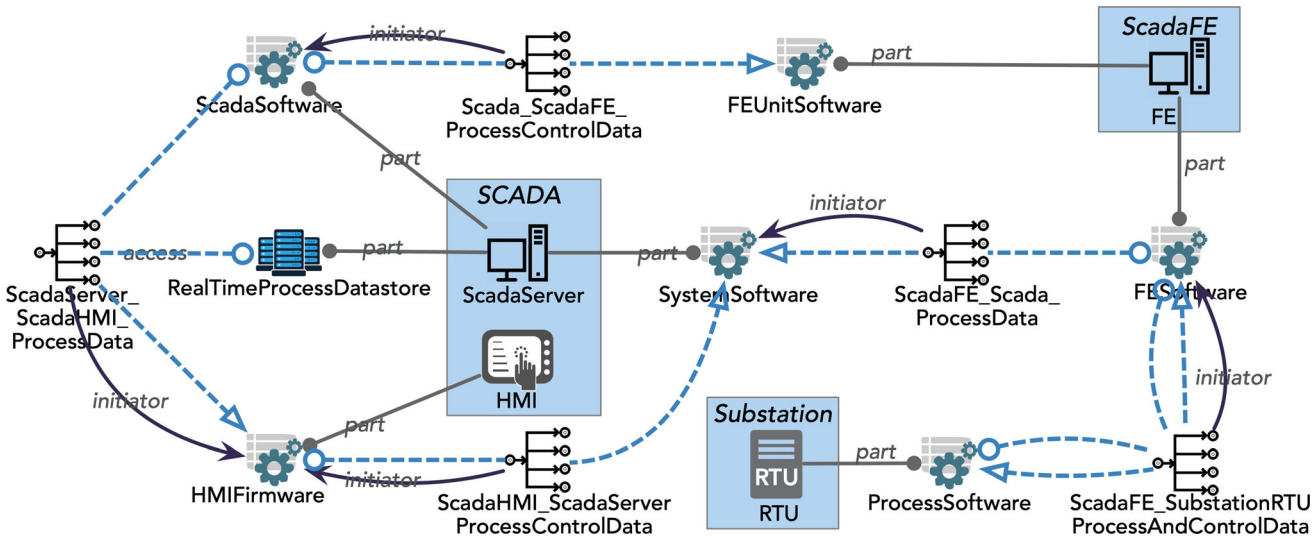


Fig. 16 Example of process data and control-commands stream. (Blue dashed arrows represent various data flows; grey arrows represent data-communication initiators.)

captured and confirmed, the safety-inspection server sends out prioritized alarms through SCADA to the SIS-PLC for emergent power-grid shut-down.

6.6.2 Historical Analysis and Load-Prediction Data

SCADA process data is inserted into the historical database with timestamps, as illustrated in Fig. 17. Historical data analysis involves several servers, namely HMI server, FTP (refers to file transfer protocol) server, historian server, and SCADA system server, to extract power generation and transmission patterns. SCADA operators query historical data from historians and visualize the data in SCADA HMIs.

6.6.3 Time-Synchronization Data

Figure 18 presents time-synchronization data organized into synchronized instances sampled from several sources and involves a range of current sensors (Fang et al. 2011). OT components such as RTUs transmit telemetry data from sensing devices to SCADA and produce time-synchronization data flows. Then, commands from the Timer server in the master supervisory system are conveyed back to the connected physical power components, to complete the

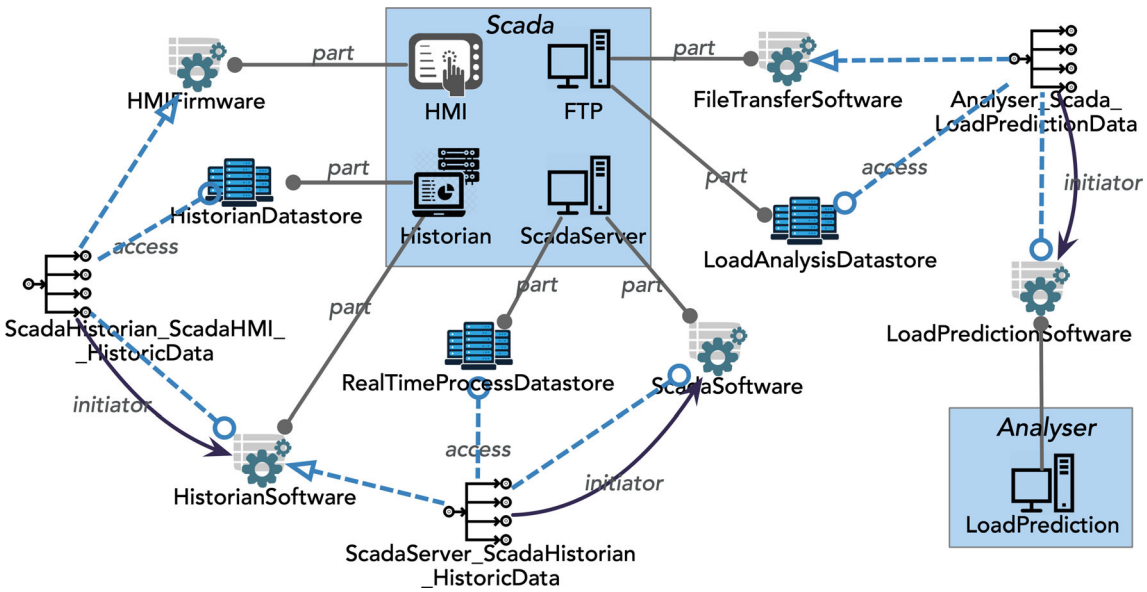


Fig. 17 Example of historical analysis and load-prediction data streams

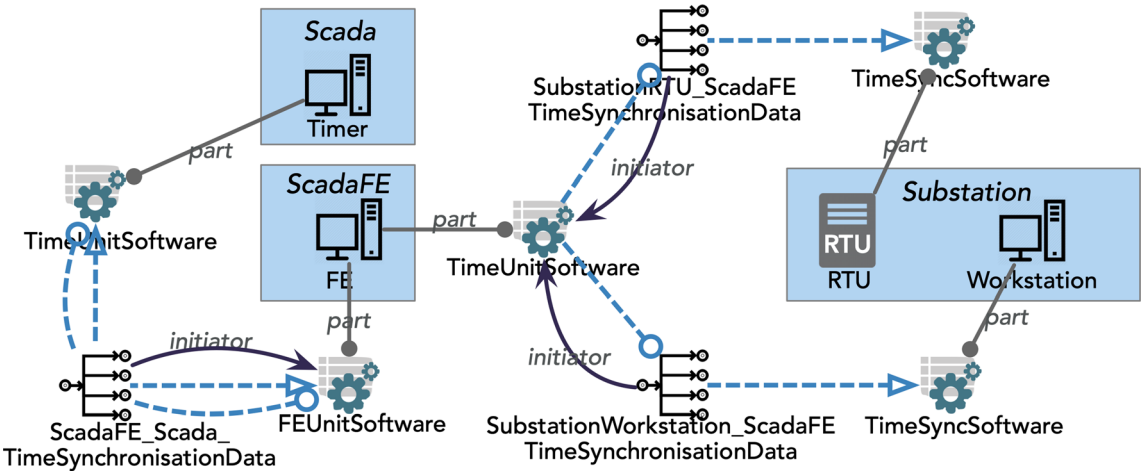


Fig. 18 Example of time-synchronization data stream

control loop process. Meanwhile, phasor measurement units (PMUs) measurements provide another common time data source for synchronization. These PMUs measure the magnitude of power signal wave (called phasors) across the smart grid, which is generally retrieved from a global positioning system (GPS) receiver (Cho et al. 2001).

6.6.4 System Update Data and Configuration Data

System updating and configuration are performed through local hosts and remote computers. Here, system update refers to updating and patching software, firmware, and operating system components. System configuration means setting up or altering parameters of RTUs, IEDs and workstations, such as input/output signal bound parameters. Here we mainly introduce remote update and configuration processes. These two data communications involve multiple stakeholders, namely software and hardware vendors, power companies, and possible outsourced IT companies. More specifically, a remote updating process starts with an update request sent out from the *Updater* hosts in *EnergySupplier* network or *Engineering* network, towards the *HWSWUpdater* host in *Vendor* network, as illustrated in Fig. 19. Software and firmware updates arrive in the *FTP* server of either *EnergySupplier* network or *Engineering* network, before being transferred to the live systems in the SCADA zone and substation zones. Subsequently, the *Updater* hosts of *Engineering* network may remotely upgrade substations such as *Substation_RTU*. The *Updater* hosts of *EnergySupplier* network can also remotely upgrade *DER* substations through *VPN* tunnels. Similarly, remote configuration of substation software or operating systems is supported by the Configuration server of either the EnergySupplier network or the Engineering network. To decrease the network exposure of directly connected hardware like *IEDs* and *SIS-PLCs*, usually, these devices are maintained or upgraded through maintenance servers and configuration databases embedded in locally connected workstations or *RTUs*.

6.6.5 Remote Login Data

Remote operators in the offices of the smart grid utility get remote access to SCADA workstations, whereby a remote login request is sent out from the remote desktop in the *LocalOffice* to the HMI server in the SCADA center, as illustrated in Fig. 20. Remote access to the local workstations is also enabled by logging into servers like *CitrixServer* from the engineering offices, usually through routers and VPNs. Sniffing software could discover the usernames and passwords of the offices, allowing attackers to build VPNs and access the control networks.

6.6.6 Web Browsing Data and Mail Data

Internet data usually includes web browsing and mail communications, commonly communicated through local office servers, as illustrated in Fig. 21. Potential vulnerabilities in web servers are improper configuration and missing authorization. Misconfiguration vulnerabilities, such as having unnecessary sample files in the web server, allow attackers to bypass authentication. Some common vulnerabilities of mail servers include weak authentication procedures and no filter for spam or phishing emails. For instance, a weak password may allow unauthorized access to a mail server, which results in information leakage.

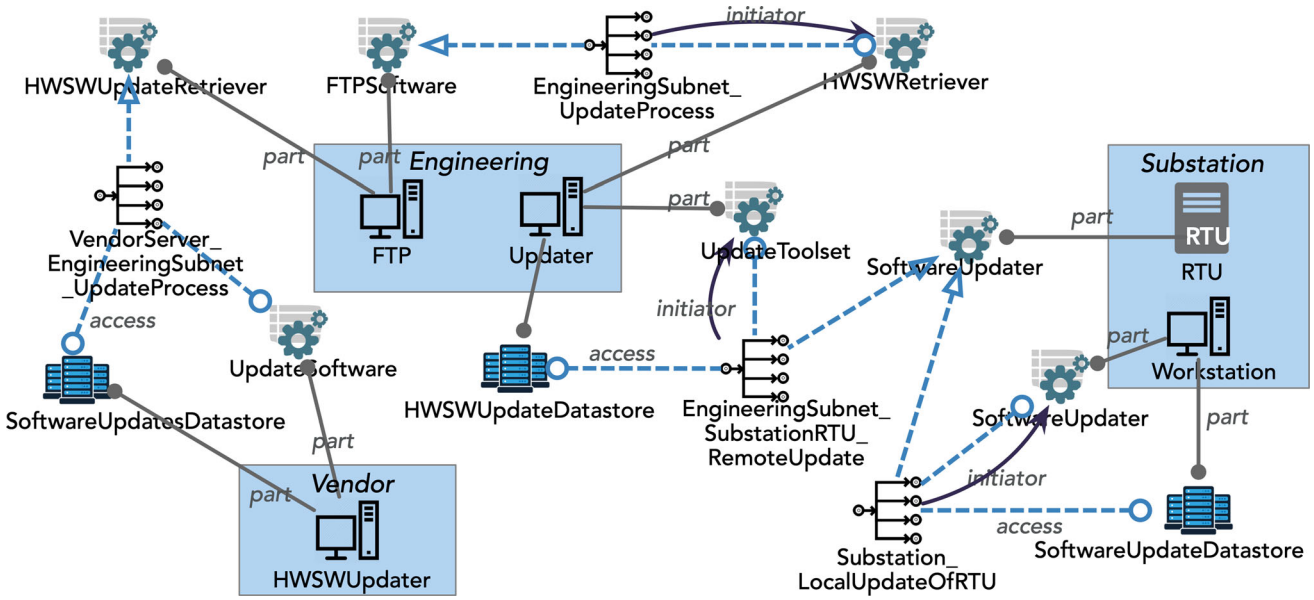


Fig. 19 Example of system updating data stream

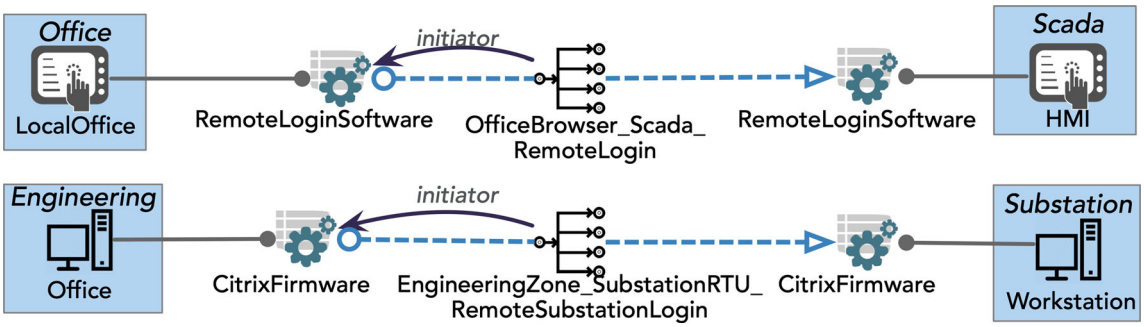


Fig. 20 Example of remote login data streams

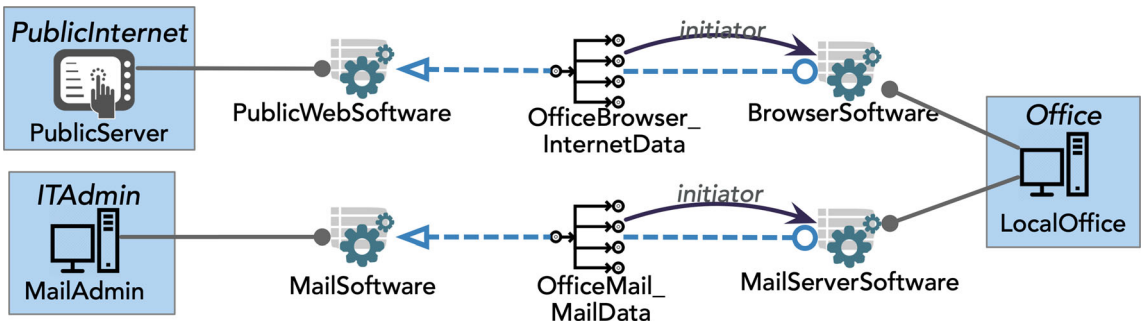


Fig. 21 Example of web browsing and mail data streams

7 Validation and Application

This section introduces the validation metrics utilized to evaluate the performances of our models, followed by a case study and result discussions.

7.1 Validation Metrics

Our evaluation process is metric based (McDaniel and Storey 2019) and follows four metrics inspired by Duque-Ramos et al. (2014), namely structural, functional adequacy, compatibility and coverage. These four metrics also cover the taxonomy development ending conditions

suggested by Nickerson et al. (2013), such as concise and explanatory requirements.

- The structural metric measures semantic models from four dimensions: (i) whether an ontology has a high cohesion with strongly related classes and a good domain coverage; (ii) whether an ontology is informative; (iii) whether an ontology provides formal relations support; and (iv) whether an ontology is related to the existence of multiple inheritances.
- The functional adequacy metric expects an ontology to have the following characteristics: (i) avoiding heterogeneous terms; (ii) providing consistent search and query; (iii) representing acquired knowledge clearly; and (iv) can be used to build other ontologies.
- The compatibility metric considers the performance of an ontology when adapted to different environments without additional actions other than those that were clarified by the ontology (i.e., adaptability).
- The coverage metric measures the range of concepts and relationships, which reflects how well the ontology represents the domain it models.

7.2 Case Study I

This case study applies our taxonomy, dependence rules and analysis method in two instantiated power-grid models. The architectures of these two models are instantiated by following the Purdue model and recommended practices for power-grid security by CISA (2022), to ensure that our models reflect power-grid structure in the real world. For the physical layer of these two models, we follow the IEEE 9-bus system that is commonly used in electricity performance analysis (Sharma et al. 2017).

7.2.1 Instantiated Power-Grid Models

We evaluate our taxonomy and rules through two instantiated power grid models *Model I* and *Model II* based on our reference models. These two models differ in terms of whether a SCADA demilitarized zone (or DMZ) is contained or not. The *Model II* example is presented in Fig. 22.

Model I contains 994 components, 1602 topological and functional dependencies, as well as 172 data flows exchanged across network applications. *Model II* contains a SCADA DMZ (Stouffer et al. 2011) as a protection layer between IT and OT networks. This DMZ zone contains replicated SCADA servers and historians. IT network can get access to the replicated historians through a firewall. *Model II* has 180 data flows. Here, we report a simplified account of the power system structure that focuses on key functionalities and connections, as illustrated in Fig. 22. We highlight some simplified examples for the same type of data flow to illustrate its function and participants.

SCADA WAN contains nine subnets that cover three primary substations (i.e., *Sub1*, *Sub4*, and *Sub5*), three secondary substations (i.e., *Sub2*, *Sub3*, and *Sub6*), and three DER (i.e., *DERSub1*, *DERSub4*, *DERSub5*). Each primary substation subsumes one RTU and one local workstation, while each secondary substation or DER substation subsumes one RTU and one mobile workstation. Here, DER substations and secondary substation are connected to *IEDs*, while primary substations are connected to both *IEDs* and *SIS-PLCs*. Substation automation is achieved through RTUs. For instance, *DERRTU4* is connected with *Bus2*, *CB1* and *CB2* to monitor and control *Generator2*.

The data flows in our instantiated *Model I* mostly follow the examples shown in Figs. 16, 17, 18, 19, 20 and 21. Some data flows in *Model II* differ. The differences between data flows in *Model I* and *Model II* are listed below.

In *Model II*, the historical process data is duplicated, transferred, and stored in a replicated historian in the SCADA DMZ FTP and be accessed to an IT network like the *Analyzer* network-zone, as illustrated in Fig. 23. Instead, historic data is accessible from *Office* network-zone in *Model I*, as shown in Fig. 17.

In *Model I*, system updating and system configuration are performed both through local hosts and remote computers, as illustrated in Fig. 19. In comparison, for *Model II*, SCADA technicians request update from hardware and software vendors, and store retrieved data in the *FTP* server of SCADA DMZ.

In *Model I*, remote engineers can directly log into local workstations through VPNs, as illustrated in Fig. 20. In *Model II*, remote operators log into the built-in HMI server in the SCADA DMZ zone to run certain HMI programming software.

7.2.2 Model-Based Dependence Analysis and Cascade Modeling

The implementing tool ConceptBase allows the declaration of specific relations via deductive rules. For example, a data connection between two components is declared once and then interpreted as a relation. The data connections between cyber components and OT devices such as RTU are of particular interest, as such data streams may be listened to or altered by malicious attackers. The following query returns such data streams:

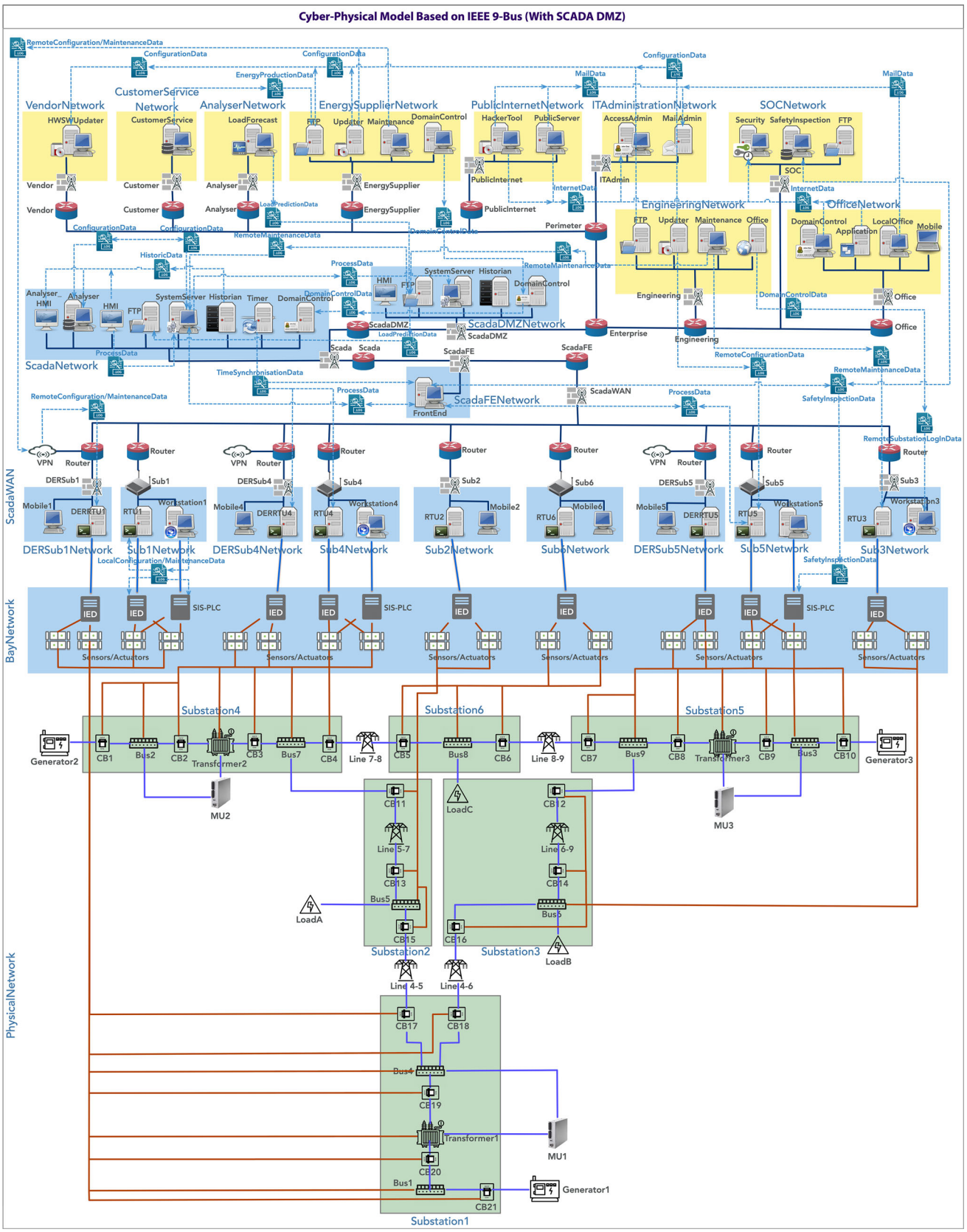


Fig. 22 Instantiated cyber-physical system based on IEEE 9Bus (with SCADA DMZ zone)

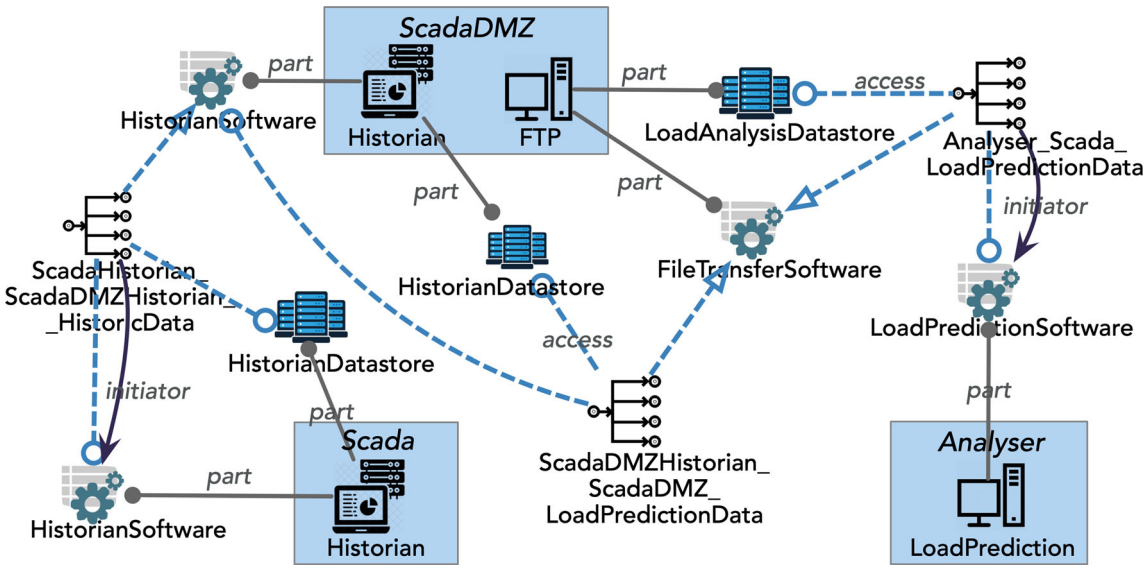


Fig. 23 Historical analysis and load-prediction data streams in *Model II* (with SCADA DMZ)

```
DatastreamToRTU in GenericQueryClass isA UnidirectionalDataStream with
   parameter
    rtu : RTU
  constraint
    cs1 : $ exists comp/Component (~rtu subcomponent comp) and  (this receiver comp) $
end
```

Using queries allows us to count the dependencies between components without following the complex network topology manually. Instead, we can automate the process of dependence calculations. Based on the defined rules in the previous Sect. 5.2, we coded our functional dependence rules in ConceptBase. We also implemented corresponding queries to extract dependencies of our instantiated models through ConceptBase, as shown below:

```
FunctionalDependentOn in QueryClass isA Component with
   computed_attribute
    functionallyDependentOn : Component
  constraint
    dfc1 : $ (this functionalDependence ~functionallyDependentOn) $
end
```

We query in our instantiated power-grid models to analyze which components functionally depend on a given node. Functional dependence is transitive. We extract six components in *Model I* that are functionally dependent on *Sub1_RTU*. We further extract multiple components that have functional dependence on these six components, which shows transitive functional dependencies. By doing so, we generate a list of dependence matrices of our targeted model. Such dependence matrices support statistical analysis or graph modeling using complex network theory. When ranking components with the highest $N_{i,j}^{FD}$ (introduced earlier in Sect. 5.2), we observe that *Model I* and *Model II* have the same top-5 components, namely

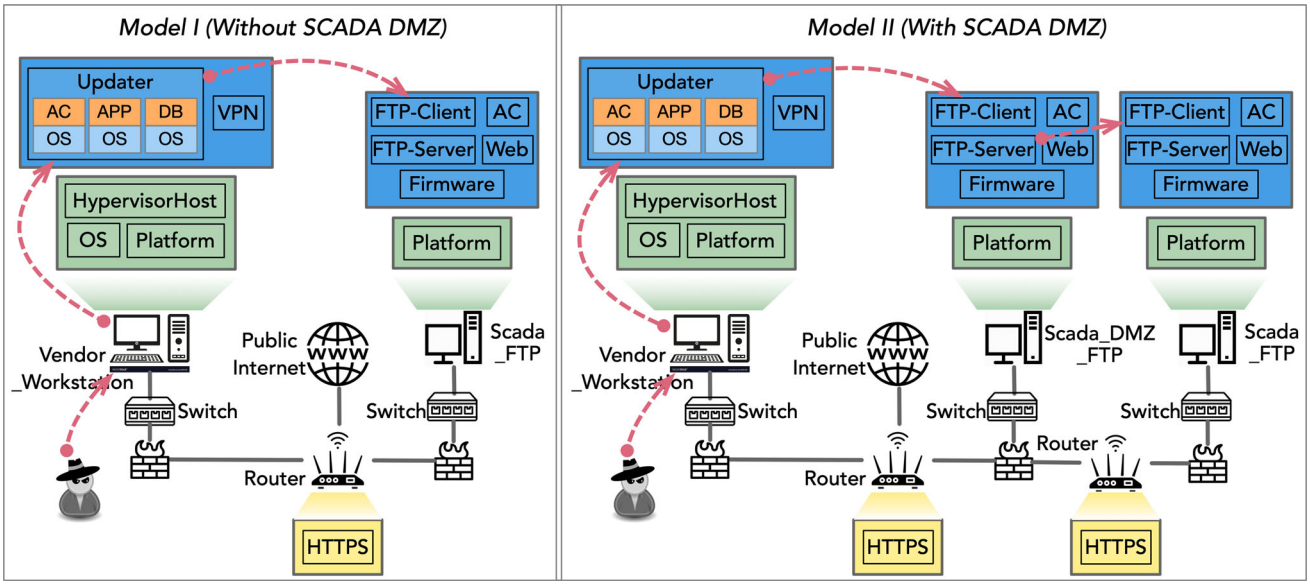


Fig. 24 Cascade failure analysis in *Model I* and *Model II*

Scada_Timer_TimeUnitSoftware ($N^{FD} = 18$), *ScadaFE_FE_FESoftware* ($N^{FD} = 10$), *Engineering_Updater_UpdateToolset* ($N^{FD} = 6$), *Engineering_Maintenance_EngineeringToolset* ($N^{FD} = 6$), and *Engineering_Updater_FWSWUpdateDatastore* ($N^{FD} = 6$).

Figure 24 illustrates two scenarios of cascading failures when setting up the same node positions with initial failures.

In the first scenario, we assume that the server *Vendor_HWSWUpdater* is compromised. In the case of *Model I*, the threat agent may further compromise the *UpdateSoftware* service, upon which false data may be injected to the data receiver like *Scada_FTP_SoftwareUpdater*, or leave a backdoor in the host. Furthermore, the threat agent may also alter the data sets in the textitSoftwareUpdatesDatastore. In the case of *Model II*, the threat agent may follow the same attack paths till the *UpdateSoftware* service. Then the threat agent needs to send a data request to the *ScadaDMZ_FTP_SoftwareUpdater*, before directly triggering a false-data injection attack.

In the second scenario, we assume that *Scada_Timer_TimeUnitSoftware* is compromised through the deployment of some existing exploits. In both *Model I* and *Model II*, compromising the time unit software in SCADA may give attackers opportunities to read further or edit time synchronization data streams between SCADA and controlling substations. This observation is in line with our conclusion earlier that nodes with higher dependencies may lead to higher importance in the context of

cybersecurity.

Moreover, the above scenarios can use the following rule and query to get what are the components that might be affected due to a failure of the starting node:

```
CascadeFailureRules in Class with
rule
rule1 : $ forall c1,c2/Component (c1 functionalDependence c2) ==> (c2 cascadeFailure c1) $
end

CascadeFailureNode in QueryClass isA Component with
   computed_attribute
    affects : Component
  constraint
    doesaffect : $ (this cascadeFailure_trans ~affects) $
end
```

For large-scale systems like smart grids, even weak adversaries could trigger cascading failures across the whole system and in the end resulting in heavy influence. One solution to improve the security of infrastructure system is to increase robustness of the system functions. Prior to developing means to measure adversarial influence or threat impacts, it is important to figure out the relationships between two components based on prior interactions. In other words, it is vital to take into consideration of the interrelated impact and composite effects when modeling and analyzing vulnerabilities in larger-scale systems. Evidence needs to be composed from all three layers. We support query vulnerabilities that exist in the system and also possible chained vulnerabilities used in APT (Chen et al. 2014). One example is illustrated earlier in Fig. 6.

7.2.3 *Case Study II Using Real-World Municipal Power Grid*

Besides the synthetic studies *Model I* and *Model II*, we also conducted a case study applying the taxonomy to the power grid of the Swedish municipality. The purpose of the case study was to validate whether the taxonomy could cover the physical and software components of a real-world power grid. The power grid consisted of two larger substations plus more than 200 smaller "transformer" stations serving neighborhoods. The two substations were identical in design. The transformer stations came in two variants, one with a single transformer, and one with two transformers.

We created a network model for the power grid components, including the OT components and networks to control the power grid components. In a second stage, we modeled the software components of the control center and, to a certain extent, the firmware on the OT components. The results of the study were as follows:

- The taxonomy could cover all components. Two new components types had to be added. One for a special balancing unit used the guarantee a common potential for the neutral power line. A second component type was added to model circuit breakers with embedded RTU. Later, we decided to use the "subcomponent" construct to model such integrated devices.
- We learned that subsystems like the transformer stations all had the same design. This led to the addition of a duplication function in the network modeler to quickly create copies of a subsystem. This applies to all internal components, including their interconnections.
- The case study revealed that the manual modeling of the power grid is rather time-consuming. Since the topology of the power grid is also stored in the SCADA system, we propose to import the model from there to minimize the manual effort and avoid errors in the manual transcription.
- The modeling of the software components led to similar conclusions about the coverage of the taxonomy. We learned that the control center heavily used virtualization, hence the hypervisor systems had to be modeled as containers of the guest operation systems, which themselves were modeled as containers of the application software.
- We did not model the data flows between the data center and the OT components because these items were not readily available.
- The information about different parts of the network model is scattered among different departments, and even the vendors of the components. This is a major challenge to create a complete and consistent network model.

Due to confidentiality requirements, the models for the real-world study were not published and were deleted after the case study was completed.

7.3 Interviews with Cybersecurity Experts

The interview-based evaluation was also performed to evaluate our tool containing the reference model discussed in Sect. 6. Each interview was designed to take between 45 and 60 min. In the first part of the interview, the system was presented using screen-dumps and slides. The second part consisted of about a dozen open-ended questions with the possibility of follow-up questions. The goal of the interviews was to evaluate the usefulness of the proposed tool in the context of a power grid operator, whose network includes both IT and OT components.

7.3.1 Background

Four interviews were conducted in a semi-structured manner in November and December, 2022, following the interview questions presented in *Appendix*. We interviewed four cybersecurity experts employed in CIs related organizations that are located in Sweden and the US.

Interviewee A is a IT security architect and consultant with more than 8 years working experience in cybersecurity of military and civilian systems. *Interviewee B* is a researcher and computer scientist who has been working on the area of threat intelligence and risk management for around 18 years. *Interviewee C* works as a IT security manager in a regional power grid company. *Interviewee D* has been responsible for IT security and digitization management in a municipal energy company for more than 3 years. Three interviewees work in organizations located in Sweden, and have in-depth knowledge of the cybersecurity needs and status of CIs, especially power grids. One interviewee works as a cybersecurity researcher in a large US based IT company.

7.3.2 Interview Results

The following text summarizes some key points obtained from the interviews, grouped by subjects. Comments not related to the subject have been omitted. When multiple people pointed out the same, the mentioned topic is included as one point.

- (i) Semantic model provides good overall picture of the system
  The proposed artifacts provide a good overall visualization of the connections and dependencies between components. Vulnerability management of a complex and large-scale IT/OT infrastructure is challenging with respects to gain a full and up-to-date overview of the vulnerability situation. Such an overview is needed as suppliers usually only provide heterogeneous documentations that are not easy to interpret. *Interviewee C* addressed that "*there are thousands of different equipment and various traffic flows that are geographically widespread in real power grids that need to be modeled and visualized. Therefore, it is important to model all types of systems based on how they are actually structured.*"Particularly, asset information is decentralized in various asset-management systems. For example, vulnerability scanner is utilized to automatically detect servers, open ports and their locations, as well as the applications embedded in these servers. Suppliers of CIs also provide some structured asset documentation that can be imported to the system inventory. Such asset information is stored "*in several different data centers*", as stated by *Interviewee C*. Nevertheless, not all components are inventorised properly. This is exemplified by quotations from *Interviewee A* that "*asset management may be outdated and may not include all details of all the relevant components.*"*Interviewee D* also commented that "*our vulnerability scanners scan the system every day. Then every once or twice a week we check the scanning results to check if anything unusual happens. We lack some kind of indication to automatically inform us of such a change. Therefore, I think this is very good to have an overview and to see which parts are more important if you are going to prioritize in some way or build away weaknesses where you see that these parts are critical and these are not so important for the operation.*"
- (ii) The artifacts bring valuable insights for vulnerability assessment
  The proposed artifacts can be used to find out which neighbor components are affected by updating a component, and to quickly look up configuration details which further aid system configuration such as configuring the firewalls to allow only the communication that is necessary according to the data flows defined in the model. The artifacts can also be used to assess redundancy such as duplicate transformers and thus the resilience of the network against disruptions. *Interviewee D* summarized that "*It is helpful to understand how to prioritize the endless amount of vulnerabilities that arise and always exist all the time.*"Although there is no standard on the frequency of IT security report generation, vulnerability trends are observed systematically. If there are few changes to the vulnerability scores, then the updates can be relatively infrequent. However, when the maximum vulnerability score of a component jumps up, an alarm should be

raised. *Interviewee B* also suggested "*hourly updates and in addition event-based alarms*"for vulnerability score update.

- (iii) The instantiated tool is helpful in integrating IT and OT cybersecurity
  Interviewee A said that "*such tool can provide a valuable service, not just for power grid companies but also for the suppliers who provide integrated solutions to the power grid companies. Suppliers could utilize such tool to demonstrate the IT security of their various offerings to their customers.*"The asset overview is distributed among multiple persons. Traditionally, different procedures are applied in managing IT and OT cybersecurity. The procedures for cybersecurity management stem from the IT side and are now gradually applied to the OT network, although OT software and firmware are usually not updated automatically like the updating process of IT software. The challenge with the OT network is that errors in patching the components lead to production breakdowns. Additionally, it is common that CI organizations out-source some IT services to data centers and specific companies. In such situations, it is harder for the CI companies to have a complete overview of the software components used for their operations. *Interviewee D* added that "*we have different documentation systems and technicians who work with different parts of the systems, but we do have an IT operating partner who has a more complete control of the cybersecurity status of the organization.*"
- (iv) Limitations of the current tool
  Interviewee A pointed out that "*one disadvantage of such a tool is the effort needed to keep it up to date.*"Besides average severity, one should also show the components with the maximum vulnerability. Different decision makers should be able to use different metrics, depending on the goal of decision making. "*Besides, one should also support measuring the proximity of a component to the Internet and its attack surface*", quoted from *Interviewee A*. *Interviewee B* gave similar suggestion that "*The metric that considers the criticality and vulnerability score is interesting as it combines the flow of commands between components with the vulnerability of the components. This metric can be normalized for further improvement. "Interviewee D* thought that besides CVSS scores, it is also important to be able to review the system protection in depth, to know what services are exposed and how, and work with the vulnerabilities in several layers, as "*some vulnerabilities with the highest CVSS base scores do not get exposed at all and less likely to be exploited, thus going down in the protection levels*".

To summarize, the proposed artifacts are useful in several perspectives, and all the interviewees expressed confidence that power grid operators would be willing to pay for a tool providing similar functions and services. The feedback from the interviewees also provide valuable guidelines on future directions to improve. For example, one interviewee pointed out that the fine-grained network model can be used to check the firewall rules based on the data and command flow definitions represented in the network model. The experts also mentioned a number of limitations of the tool. We regard this not a critique of the tool but rather as a fair assessment of its specific functions. Overall, the experts expressed that the tool fills a gap by a fined-grained integrated enterprise model for power grid operators that allows to decorate the included software components by their aggregated vulnerability scores and to combine this information with the context of these components in the overall enterprise model. The experts concluded that such a tool is useful and not yet available in this form in the market.

7.4 Discussion

In this section, we compare our method with two similar approaches, one is based on ArchiMate (Lankhorst et al. 2010), and the other is named powerLang (Hacks et al. 2020) that is developed from MAL (meta attack language, Johnson et al. (2018)) and CySeMoL (Sommestad et al. 2013). The idea of a comprehensive model that includes all relevant dependencies is crucial to understanding the effect of cyber-attacks on software applications and system software on the ability of the enterprise to perform its business processes. While cybersecurity was not the original purpose of enterprise modeling frameworks, Archimate is utilized to support cybersecurity management in several studies (Grandry et al. 2013; Hacks et al. 2019). ArchiMate does support some form of system decomposition, e.g., a server computer contains the operating system and application software running on them. Efforts to include security aspects into ArchiMate (Ellerm and Morales-Trujillo 2020) are predominantly focusing on design rather than analysis of vulnerabilities. ArchiMate is not (yet) designed to cover the plethora of OT and physical components found in CIs, such as smart grids. ArchiMate does not support modeling of power flow either. powerLang (Hacks et al. 2020) provides a meta model that is built on top of MAL to support automated attack-analysis purpose in the power domain. powerLang (for now) covers only general aspects of IT and OT assets which are not enough to model cascading effects of these different assets when exploited, and is not (yet) evaluated in a real-world case setting.

Our power-grid taxonomy covers standardized virtual replicas for cyber connections, cyber-physical setup, and physical processes. Relations between different classes cover sub-component configuration, data connection, and power connection, and can be easily extended, which shows a high structural strength of our model. We further define the functional dependence rules to formally assess the levels of connections. These dependence rules reuse names from existing reliable ontology and frameworks to ensure conciseness. We show how the taxonomy can be implemented in a power grid prototype. Such instantiated power-grid models can be leveraged to perform dependence identification and cascade modeling, thus supporting security analysis. Actually, the presentation of our proposed taxonomy and reference model uses ConceptBase for its ability to represent both classes and objects in the same database. This allows us to use the taxonomy as constructs of a domain-specific modeling language to represent sample smart grids to any degree of detail. We visualize the security topology of the smart grid before and after an attack and answer questions like which IT and OT components are affected by the attack and how the attack propagated throughout the network, in *Case study I*. We also carried out *Case study II* to validate the functional adequacy and coverage of our model in a real-world setting. The presented case studies show the strength of our model in terms of consistent search and query support, as well as knowledge reuse. For example, the reference models can be used in power-grid modeling, presenting high functional adequacy. The real-world case study revealed that more automation in creating integrated models is needed. Much of the information is available from the management software itself. The coverage of the taxonomy was sufficient. The taxonomy had to be extended on the fly on a few occasions, supported by the ConceptBase system since it manages the network models and the taxonomy in a single database. We evaluated further the utility of our artifacts by performing four interviews with cybersecurity experts in CIs. We received confirmations on the usefulness of our proposed methods in supporting dependence analysis and vulnerability analysis of complex systems, and also collected notes on the limitations of our methods from the application users' perspectives.

There are some limitations of our current model. Firstly, our function dependence rule set is not complete in terms of all possible relations between components in the cyber and cyber-physical layer. Yet still, our dependence rules are multi-dimensional, and serve as basis for further extension. For example, we plan to further differentiate the dependence levels of different data streams, such as controlling data and processing data. Secondly, the implemented CVSS mechanisms for vulnerability-severity calculation are not fully fit with the industrial requirement and environment. This limitation can be addressed by integrating other severity mechanisms or adjusting the existing mechanisms with weights suggested by CI operators. Thirdly, the results of the vulnerability-assessment method depend on the data quality of collected CI system configuration and the open-accessible vulnerability repositories. Fourthly, our model do not (yet) support business service layer, which can be easily extended with the current system.

8 Conclusion

This paper delivers a modeling methodology of intricate critical infrastructure networks, and related constraints via a taxonomy and reference models. These two modules can also serve as a knowledge base of IT/OT convergent CI models that are analyzed by external tools for vulnerability analysis. Current CI vulnerability management is challenging due to the knowledge gap between IT security and OT security, and also different terminologies used in these two domains. Our model bridges such gaps with common semantics, and supports query of vulnerabilities across the CI layers. Static analysis queries are used to pinpoint design weaknesses in the layered network of CIs. Using the proposed CI reference models (power-grid models in this paper) provides disciplined and coherent support to specify and group components and coordination mechanisms as a mean to harness the notorious complexity of CI networks. We also define multi-dimensional cyber and cyber-physical functional dependencies that support cascade modeling and component-criticality analysis, particularly depending on their role in controlling the physical process, e.g., electric power delivery.

This paper also presents a vulnerability assessment method that integrates the proposed taxonomy and functional dependence rules, while gathering vulnerability instances from repositories such as NVD for targeted components, to support vulnerability analytics of the investigated system. Details of the vulnerability gathering and correlation process can be found in the authors' previous work (reference hidden). In doing so, multiple data sources can be correlated to support further automated architecture modeling.

We instantiated our model in ConceptBase that implemented Telos language using a proposition (equivalent to object) data structure. We defined deductive rules and queries that are used to propagate properties such as nominal voltage and frequency of physical power-grid components. We built upon and reused terms in existing ontologies, and followed reliable frameworks such as the Purdue model, NIST SP 800-82, and the IEC 62351 series

during the design and development of our taxonomy, dependence rules, and reference models to enhance the compatibility strength of our works. The contributions of this paper are summarized earlier in Sect. 1.3.

We conducted a three-fold validation of our approach. First, we developed a reference model for a power grid company based on the IEEE Nine-Bus model. The reference model validated that a comprehensive enterprise model including the physical layer of the power-grid can be represented. Second, we applied our approach to a municipal power grid company. This confirmed that the real-world enterprise model can be created with our tool and vulnerabilities of the described software components can be looked-up from public repositories using the enterprise model. Thirdly, we conducted four semi-structured interviews with domain experts to validate the usefulness of our approach based on the work experience of the domain experts. Two domain experts were IT security experts from municipal power-grid companies. The other two were IT security experts from a consulting company and a network solution provider, respectively. The interviews confirmed the usefulness of the integrated enterprise model that combines IT and OT aspects of the enterprise, which are typically managed by different departments and not integrated in the current practice of municipal power-grid companies. Further, the interviews confirmed that the automatic update of vulnerability details attached to the enterprise model is of great value to IT experts in the companies, as it saves manual error-prone updates.

Future work directions include extension of reference models with more predefined modules and settings of component types, as well as further expansion of cyber-physical dependencies that contribute to system reliability and robustness analysis with some graph-mining techniques. Weighted probability would be one of the options in cascade modeling. For example, we plan to investigate possibility of a failure caused by a security incident, namely, the possibility for a system to fail if the connected component is already compromised.

We also plan to include the business layer by modeling the business processes of smart grid companies, such as trading, maintenance and customer management. We did not cover it in this paper since this level is rather well-understood. The power-grid industry is evolving to support smart services to their customers, such as home charging of electric vehicles and energy management for smart homes. This will lead to an order of magnitude more complex models to assess the cybersecurity of the power-grid. Another trend in the industry is outsourcing, in particular of the IT. This will create new challenges to maintain the integrated models needed to assess the cybersecurity. We are currently investigating how to model import interfaces that honor confidentiality of information across company boundaries.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12599-023-00811-0.

Acknowledgements We thank the colleagues from the ELVIRA project for their contributions to earlier versions of the taxonomy. We are in particular grateful to Yacine Atif for his support and encouragement. Many thanks also to the interview partners for helping to validate the usefulness of our approach. Finally, we thanks the anonymous reviewers for their diligent and constructive evaluations.

Funding Open access funding provided by University of Skövde.

References

ABB (2022) ABB energy manager. https://new.abb.com/industrial-software/sustainability/energy-manager/industrial-energy-load-planning-forecasting-scheduling, Accessed 23 April 2022

Abubakar I, Khalid S, Mustafa M, Shareef H, Mustapha M (2017) Application of load monitoring in appliances' energy management-a review. Renew Sustain Energ Rev 67:235–245

Akbarzadeh A, Katsikas S (2021) Identifying and analyzing dependencies in and among complex cyber physical systems. Sens 21(5):1685

Alcaraz C (2019) Secure interconnection of IT-OT networks in industry 4.0. Critical infrastructure security and resilience. Springer, Heidelberg, pp 201–217

Bernstein PA, Haas LM (2008) Information integration in the enterprise. Commun ACM 51(9):72–79

Bhamare D, Zolanvari M, Erbad A, Jain R, Khan K, Meskin N (2020) Cybersecurity for industrial control systems: a survey. Comput Secur 89(101):677

Blockley D, Agarwal J, Pinto J, Woodman N (2002) Structural vulnerability, reliability and risk. Prog Struct Eng Mater 4(2):203–212

Boyer SA (2009) SCADA: supervisory control and data acquisition. International Society of Automation, Pittsburgh

Boyes H, Hallaq B, Cunningham J, Watson T (2018) The industrial internet of things (IIoT): an analysis framework. Comput Ind 101:1–12

Brand K, Brunner C, Wimmer W (2011) Design of IEC 61850 based substation automation systems according to customer requirements. Indian J Power River Val Dev 61(5):87

Brand KP, Wimmer W, Lohmann V (2003) Substation automation handbook. Utility Automation Consulting Lohmann Bremgarten, Switzerland

Burkett JS (2012) Business security architecture: weaving information security into your organization's enterprise architecture through sabsa®. Inf Secur J Glob Perspect 21(1):47–54. https://doi.org/10.1080/19393555.2011.629341

Bytschkow D, Campetelli A, Cengarle MV, Irlbeck M, Schorp K (2014) Reference framework for the engineering of cyber-physical systems: a first approach. TU München. https://mediatum.ub.tum.de/1197504

Cheminod M, Durante L, Valenzano A (2012) Review of security issues in industrial networks. IEEE Trans Ind Inform 9(1):277–293

Chen P, Desmet L, Huygens C (2014) A study on advanced persistent threats. In: IFIP international conference on communications and multimedia security. Springer, Heidelberg, pp 63–72

Cho KS, Shin JR, Hyun SH (2001) Optimal placement of phasor measurement units with GPS receiver. In: 2001 IEEE power engineering society winter meeting. Conference proceedings (cat. no. 01ch37194), IEEE, vol 1, pp 258–262

Chopade P, Bikdash M (2011) Critical infrastructure interdependency modeling: using graph models to assess the vulnerability of smart power grid and scada networks. In: 2011 8th international conference & expo on emerging technologies for a smarter world, IEEE, pp 1–6

CISA (2022) Cybersecurity & infrastructure security agency. https://www.cisa.gov/uscert/ics/Recommended-Practices, Accessed 23 April 2022

Cloutier R, Muller G, Verma D, Nilchiani R, Hole E, Bone M (2010) The concept of reference architectures. Syst Eng 13(1):14–27

Conklin WA (2016) IT vs. OT security: a time to consider a change in CIA to include resilienc. In: 2016 49th Hawaii international conference on system sciences (HICSS), IEEE, pp 2642–2647

Diefenbach T, Lucke C, Lechner U (2019) Towards an integration of information security management, risk management and enterprise architecture management – a literature review. In: 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Sydney, Australia, December 11-13, 2019, IEEE, pp 326–333

Disterer G (2013) The concept of reference architectures. J Inf Secur. https://doi.org/10.4236/jis.2013.42011

Duque-Ramos A, Boeker M, Jansen L, Schulz S, Iniesta M, Fernández-Breis JT (2014) Evaluating the good ontology design guideline (GoodOD) with the ontology quality requirements and evaluation method and metrics (OQuaRE). PloS One 9(8):104463

Eckhart M, Ekelhart A (2018) Towards security-aware virtual environments for digital twins. In: Proceedings of the 4th ACM workshop on cyber-physical system security, pp 61–72

Ekstedt M, Sommestad T (2009) Enterprise architecture models for cyber security analysis. In: 2009 IEEE/PES power systems conference and exposition, pp 1–6, https://doi.org/10.1109/PSCE.2009.4840267

Ellerm A, Morales-Trujillo ME (2020) Modelling security aspects with archimate: a systematic mapping study. In: 46th euromicro conference on software engineering and advanced applications, SEAA 2020, Portoroz, Slovenia, Aug 26-28, IEEE, pp 577–584

Falliere N, Murchu LO, Chien E (2011) W32. Stuxnet dossier. White paper, Symantec Corp, Secur Response 5(6):29

Fang X, Misra S, Xue G, Yang D (2011) Smart grid - the new and improved power grid: a survey. IEEE Commun Surv Tutor 14(4):944–980

Feiler PH, Lewis B, Vestal S (2003) The SAE Avionics Architecture Description Language (AADL) standard: A basis for model-based architecture-driven embedded systems engineering. Tech. rep., Army Aviation and Missile Command Redstone Arsenal AL. https://apps.dtic.mil/sti/citations/ADA612735

FIRST (2022) Common vulnerability scoring system. https://www.first.org/cvss/, Accessed 23 April 2022

Fredriksen R, Kristiansen M, Gran BA, Stølen K, Opperud TA, Dimitrakos T (2002) The CORAS framework for a model-based risk management process. In: International conference on computer safety, reliability, and security. Springer, Heidelberg, pp 94–105

Gottschalk M, Uslar M, Delfs C (2017) The use case and smart grid architecture model approach: the IEC 62559–2 use case template and the SGAM applied in various domains. Springer, Heidelberg

Grandry E, Feltus C, Dubois E (2013) Conceptual integration of enterprise architecture management and security risk management. In: Bagheri E, Gasevic D, Hallé S, Hatala M, Nezhad HRM, Reichert M (eds) 17th IEEE international enterprise distributed object computing conference workshops, EDOC workshops, Vancouver, BC, Canada, Sept 9-13, 2013, IEEE Computer Society, pp 114–123

Guo H, Zheng C, Iu HHC, Fernando T (2017) A critical review of cascading failure analysis and modeling of power system. Renew Sustain Energy Rev 80:9–22

Hacks S, Hacks A, Katsikeas S, Klaer B, Lagerström R (2019) Creating meta attack language instances using archimate: applied to electric power and energy system cases. In: 2019 IEEE 23rd international enterprise distributed object computing conference (EDOC), IEEE, pp 88–97

Hacks S, Katsikeas S, Ling E, Lagerström R, Ekstedt M (2020) PowerLang: a probabilistic attack simulation language for the power domain. Energy Inf 3(1):1–17

He H, Yan J (2016) Cyber-physical attacks and defences in the smart grid: a survey. IET Cyber-Phys Syst Theory Appl 1(1):13–27

Humayed A, Lin J, Li F, Luo B (2017) Cyber-physical systems security - a survey. IEEE Internet Things J 4(6):1802–1831

IEEE (2011) IEEE guide for smart grid interoperability of energy technology and information technology operation with the electric power system (EPS), end-use applications, and loads. IEEE, New York

Irlbeck M, Bytschkow D, Hackenberg G, Koutsoumpas V (2013) Towards a bottom-up development of reference architectures for smart energy systems. In: 2013 2nd international workshop on software engineering challenges for the smart grid (SE4SG), IEEE, pp 9–16

Janulevičius J, Marozas L, Čenys A, Goranin N, Ramanauskaité S (2017) Enterprise architecture modeling based on cloud computing security ontology as a reference model. In: 2017 open conference of electrical, electronic and information sciences (eStream), pp 1–6, https://doi.org/10.1109/eStream.2017.7950320

Jarke M, Gallersdörfer R, Jeusfeld MA, Staudt M (1995) ConceptBase - a deductive object base for meta data management. J Intell Inf Syst 4(2):167–192

Jeusfeld M, Jarke M, Mylopoulos J (2009) Metamodeling for method engineering. MIT Press, Cambridge

Johnson P, Lagerström R, Ekstedt M (2018) A meta language for threat modeling and attack simulations. In: Proceedings of the 13th international conference on availability, reliability and security, pp 1–8

Kandias M, Mylonas A, Theoharidou M, Gritzalis D (2011) Exploitation of auctions for outsourcing security-critical projects. In: 2011 IEEE symposium on computers and communications (ISCC), IEEE, pp 646–651

Khan R, McLaughlin K, Laverty D, Sezer S (2017) Stride-based threat modeling for cyber-physical systems. In: 2017 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe), IEEE, pp 1–6

Knapp ED, Langill JT (2014) Industrial network security: securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems. Syngress, Oxford

Knapp ED, Samani R (2013) Applied cyber security and the smart grid: implementing security controls into the modern power infrastructure. Elsevier, Amsterdam

Kong PY (2019) Optimal configuration of interdependence between communication network and power grid. IEEE Trans Ind Inf 15(7):4054–4065

König S, Rass S, Rainer B, Schauer S (2019) Hybrid dependencies between cyber and physical systems. In: Intelligent computing-proceedings of the computing conference, Springer, Heidelberg, pp 550–565

Korman M, Lagerström R, Välja M, Ekstedt M, Blom R (2016) Technology management through architecture reference models: a smart metering case. In: 2016 Portland international conference on management of engineering and technology (PICMET), IEEE, pp 2338–2350

Kure H, Islam S, Razzaque M (2018) An integrated cyber security risk management approach for a cyber-physical system. Appl Sci 8(6):898

Kwasinski A (2020) Modeling of cyber-physical intra-dependencies in electric power grids and their effect on resilience. In: 2020 8th workshop on modeling and simulation of cyber-physical energy systems, IEEE, pp 1–6

Lallie HS, Debattista K, Bal J (2018) An empirical evaluation of the effectiveness of attack graphs and fault trees in cyber-attack perception. IEEE Trans Inf Forensics Secur 13(5):1110–1122

Lankhorst MM, Proper HA, Jonkers H (2010) The anatomy of the ArchiMate language. Int J Inf Syst Model Des 1(1):1–32

Leune K, Kim S (2021) Supporting cyber threat analysis with service-oriented enterprise modeling. In: di Vimercati SDC, Samarati P (eds) Proceedings of the 18th international conference on security and cryptography (SECRYPT), July 6-8, Scitepress, pp 385–394

Liu L, Eric S, Mylopoulos J (2009) Secure-i*: engineering secure software systems through social analysis. Int J Softw Inf 3(1):89–120

Marashi K, Sarvestani SS, Hurson AR (2017) Consideration of cyber-physical interdependencies in reliability modeling of smart grids. IEEE Trans Sustain Comput 3(2):73–83

McDaniel M, Storey VC (2019) Evaluating domain ontologies: clarification, classification, and challenges. ACM Comput Surv (CSUR) 52(4):1–44

MITRE (2021) Cve-2021-36745. https://nvd.nist.gov/vuln/detail/CVE-2021-36745, Accessed 23 April 2022

MITRE (2022a) Common attack pattern enumeration and classification. https://capec.mitre.org/index.html, Accessed 23 April 2022

MITRE (2022b) Common platform enumeration. https://cpe.mitre.org/, Accessed 23 April 2022

MITRE (2022c) Common vulnerability enumeration. https://cve.mitre.org/, Accessed 23 April 2022

MITRE (2022d) Common weakness enumeration. https://cwe.mitre.org/index.html, Accessed 23 April 2022

Mitsubishi Electric (2022) Melsec-q plc. https://www.mitsubishielectric.com/fa/products/cnt/plcq/items/index.html, Accessed 23 April 2022

Mo Y, Kim THJ, Brancik K, Dickinson D, Lee H, Perrig A, Sinopoli B (2011) Cyber-physical security of a smart grid infrastructure. Proc IEEE 100(1):195–209

Mohamed MA, Kardas G, Challenger M (2021) Model-driven engineering tools and languages for cyber-physical systems - a systematic literature review. IEEE Access 9:48605–48630

Mohurle S, Patil M (2017) A brief study of wannacry threat: ransomware attack 2017. Int J Adv Res Comput Sci 8(5):1938–1940

Mouratidis H, Giorgini P (2007) Secure tropos: a security-oriented extension of the tropos methodology. Int J Softw Eng Knowl Eng 17(02):285–309

Mozzaquatro BA, Melo R, Agostinho C, Jardim-Goncalves R (2016) An ontology-based security framework for decision-making in industrial systems. In: 2016 4th international conference on model-driven engineering and software development (MODELSWARD), IEEE, pp 779–788

Mozzaquatro BA, Agostinho C, Goncalves D, Martins J, Jardim-Goncalves R (2018) An ontology-based cybersecurity framework for the internet of things. Sens 18(9):3053

Murray G, Johnstone MN, Valli C (2017) The convergence of it and 2141 OT in critical infrastructure. In: Proceedings of 15th Australian Information Security Management Conference, pp 149–155

Myhre SF, Fosso OB, Heegaard PE, Gjerde O, Kjølle GH (2020) Modeling interdependencies with complex network theory in a combined electrical power and ICT system. In: 2020 international conference on probabilistic methods applied to power systems (PMAPS), IEEE, pp 1–6

Mylopoulos J, Borgida A, Jarke M, Koubarakis M (1990) Telos: representing knowledge about information systems. ACM Trans Inf Syst (TOIS) 8(4):325–362

NERC (2008) North american electric reliability corporation (NERC) critical infrastructure protection (CIP). https://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx, Accessed 23 April 2022

Nguyen PH, Ali S, Yue T (2017) Model-based security engineering for cyber-physical systems: a systematic mapping study. Inf Softw Technol 83:116–135

Nickerson RC, Varshney U, Muntermann J (2013) A method for taxonomy development and its application in information systems. Eur J Inf Syst 22(3):336–359

NIST (2014) Framework for improving critical infrastructure cybersecurity. https://doi.org/10.6028/NIST.CSWP.02122014, Accessed 23 April 2022

NIST (2022) National vulnerability database. https://nvd.nist.gov/vuln, Accessed 23 April 2022

Noel S, Harley E, Tam K, Limiero M, Share M (2016) Cygraph: graph-based analytics and visualization for cybersecurity. Handbook of statistics. vol 35. Elsevier, Amsterdam, pp 117–167

Oliva GA, Santana FW, Gerosa MA, De Souza CR (2011) Towards a classification of logical dependencies origins: a case study. In: Proceedings of the 12th international workshop on principles of software evolution and the 7th annual ERCIM workshop on software evolution, pp 31–40

Ouyang M (2014) Review on modeling and simulation of interdependent critical infrastructure systems. Reliab Eng Syst Saf 121:43–60

Palm J (2021) Exploring limited capacity in the grid: actors, problems, and solutions. Front Energy Res 9:199

Pavleska T, Aranha H, Masi M, Grandry E, Sellitto GP (2019) Cybersecurity evaluation of enterprise architectures: The e-sens case. In: Gordijn J, Guédria W, Proper HA (eds) The practice of enterprise modeling – 12th IFIP working conference, PoEM 2019, Luxembourg, Nov 27-29, 2019, proceedings, Springer, Lecture Notes in Business Information Processing, vol 369, pp 226–241

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24(3):45–77

PES I (2008) IEEE standard for SCADA and automation systems. vol IEEE Std C 37

Ruland KC, Sassmannshausen J, Waedt K, Zivic N (2017) Smart grid security - an overview of standards and guidelines. e & i Elektrotech Inf 134(1):19–25

Scheer A, Nüttgens M (2000) ARIS architecture and reference models for business process management. In: van der Aalst WMP, Desel J, Oberweis A (eds) Business process management, models, techniques, and empirical studies, Springer, Heidelberg, Lecture Notes in Computer Science, vol 1806, pp 376–389

Schiffman M (2011) The common vulnerability reporting framework. An Internet Consortium for Advancement of Security on the Internet (ICASI), Whitepaper, Version 1

SEGRID Consortium (2017) Security for smart electricity grids, how to address the security challenges in smart grids. Tech. rep., Segrid.eu, https://segrid.eu/wp-content/uploads/2017/10/White paper-Segrid-9-FV.pdf, Accessed 23 April 2022

Sharma S, Velgapudi NS, Pandey K (2017) Performance analysis of IEEE 9 bus system using TCSC. In: 2017 recent developments in control, automation & power engineering (RDCAPE), IEEE, pp 251–256

Shepard M (2015) Getting started with powershell. Packt Publishing Ltd, Birmingham

Sommestad T, Ekstedt M, Holm H (2013) The cyber security modeling language: a tool for assessing the vulnerability of enterprise system architectures. IEEE Syst J 7(3):363–373

Stouffer K, Falco J, Scarfone K et al (2011) Guide to industrial control systems (ICS) security. NIST Spec Publ 800(82):16–16

Suryn W, Abran A, April A (2003) ISO/IEC SQuaRE: the second generation of standards for software product quality. http://publicationslist.org/data/a.april/ref-182/Suryn,%20Abran,%20April.pdf

Uslar M, Rohjans S, Neureiter C, Pröstl Andrén F, Velasquez J, Steinbrink C, Efthymiou V, Migliavacca G, Horsmanheimo S, Brunner H et al (2019) Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: a European perspective. Energy 12(2):258

Vaiman M, Bell K, Chen Y, Chowdhury B, Dobson I, Hines P, Papic M, Miller S, Zhang P (2012) Risk assessment of cascading outages: methodologies and challenges. IEEE Trans Power Syst 27(2):631

Venkata RY, Kamongi P, Kavi K (2018) An ontology-driven framework for security and resiliency in cyber physical systems. ICSEA 2018:23

Vielberth M, Böhm F, Fichtinger I, Pernul G (2020) Security operations center: a systematic study and open challenges. IEEE Access 8:227756–227779

Wang C, Xing L, Levitin G (2012) Competing failure analysis in phased-mission systems with functional dependence in one of phases. Reliab Eng Syst Saf 108:90–99

Webster J, Watson RT (2002) Analyzing the past to prepare for the future: writing a literature review. MIS Q 26(2):xiii–xxiii

Whitehead DE, Owens K, Gammel D, Smith J (2017) Ukraine cyber-induced power outage: analysis and practical mitigation strategies. In: 2017 70th annual conference for protective relay engineers (CPRE), IEEE, pp 1–8

Williams TJ (1994) The purdue enterprise reference architecture. Comput Ind 24(2–3):141–158

Xu LD, Xu EL, Li L (2018) Industry 4.0: state of the art and future trends. Int J Prod Res 56(8):2941–2962

Yin RK (2009) Case study research: design and methods, vol 5. Sage, Thousand Oaks

Ying Z, Yirong W, Ning W (2014) Study of network architecture and ip address allocation of wireless VPN for power grid. In: 2014 enterprise systems conference, IEEE, pp 305–309

Zeinali M, Thompson J (2021) Comprehensive practical evaluation of wired and wireless internet base smart grid communication. IET Smart Grid 4(5):522–535

Zhao G, Xing L (2019) Competing failure analysis considering cascading functional dependence and random failure propagation time. Qual Reliab Eng Int 35(7):2327–2342

Zhou Q, Natarajan S, Simmhan Y, Prasanna V (2012) Semantic information modeling for emerging applications in smart grid. In: Information technology: New generations (ITNG), 2012 ninth international conference on, IEEE, pp 775–782

Zhu W, Milanović JV (2017) Interdepedency modeling of cyber-physical systems using a weighted complex network approach. In: 2017 IEEE Manchester Powertech, IEEE, pp 1–6