This is a post-print of the original paper in Proc. 22nd IEEE Enterprise Distributed Object Computing Conference (EDOC 2018), Stockholm, Sweden, Oct. 16-19, 2018;

A Language and Repository for Cyber Security of Smart Grids

Yuning Jiang, Manfred Jeusfeld, Yacine Atif, Jianguo Ding
School of Informatics
University of Skövde, Sweden
{firstname.lastname}@his.se

Christoffer Brax, Eva Nero
Combitech AB
Skövde, Sweden
{firstname.lastname}@combitech.se

***Abstract*—Power grids form the central critical infrastructure in all developed economies. Disruptions of power supply can cause major effects on the economy and the livelihood of citizens. At the same time, power grids are being targeted by sophisticated cyber attacks. To counter these threats, we propose a domain-specific language and a repository to represent power grids and related IT components that control the power grid. We apply our tool to a standard example used in the literature to assess its expressiveness.**

***Keywords*-cyber security, enterprise architecture, domain-specific language, taxonomy**

I. Introduction

The Smart Grid language and repository aim at categorizing the list of involved components and splitting them into a set of meaningful concepts and relationships. The proposed smart grid elements are distinguished according to their types, namely power and cyber elements. This distinction is used for visual arrangement of physical and control components and their event-driven interaction. The smart grid taxonomy is intended for individuals seeking an overview of smart grids in terms of inventory and valid relationships in the process of enhancing security attributes or investigating threat scenarios. It also enhances correctness and productivity of smart grid model specification by using well-defined concepts and relationships. This paper proposes a taxonomy framework for the specification of smart-grid architecture instances. The framework includes a specification language for interaction logics and component architectural-diagrams.

The definitions assign meaning to the cyber-physical components as well as subsystems of smart grids and their dependencies. As a multidisciplinary area, the components and subsystems in smart grids need to obey different levels of rules. For example, a generation substation needs to obey not just the electricity generation theories, e.g. Faradays law, but also the data exchange protocols and rules for connecting physical and cyber components. The taxonomy supports the proper description of interconnections between both cyber and physical parts. We contribute a modeling methodology of intricate smart-grid networks, and related constraints via taxonomy structure. We designed a domain-specific language which orchestrates component interactions across the taxonomy. Using the proposed smart-grid taxonomy provides a disciplined and coherent support to specify and group components and coordination mechanisms as a mean to harness the notorious complexity of smart-grid networks.

*This research has been supported in part by the EU ISF Project A431.678/2016 ELVIRA (Threat modeling and resilience of critical infrastructures), coordinated by Polismyndigheten/Sweden.

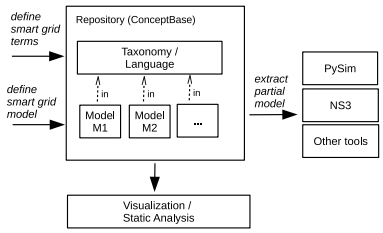

Figure 1. Repository architecture for smart grid models

Figure 1 shows the overall architecture and purpose of the repository for smart grids. The upper level of the repository contains the taxonomy of all smart grid components relevant for assessing the smart grid vulnerability. These terms are the classes for actual components occurring in smart grid models such as model M1, M2 and so forth. The models cover both the power grid topology as well as the control and monitoring topology. The analysis tools on right side of Figure 1 extract the static smart grid topology from the repository. Power simulation tools would only extract the power grid part of the data, while network simulation tools would utilize the communication data. Further, the repository provides services to visualize and statically analyze smart grid model, e.g. to discover model errors.

Ontologies are used as a vocabulary basis consisting of facts (both abstract facts and entity facts), constraints, types and attributes of different agents in the smart grids. The multi-agent system of [1] includes one control agent, one distributed energy resources (DER) agent, one user agent and one database agent. A second purpose of ontologies is to support the integration of semantic information integration from heterogeneous information data and distribution networks, such as smart meter data and distributed social networks. Zhou et al. state the importance of applying semantic information for knowledge base construction and information reuse [2]. In addition, the semantic information can be used for data mining and rule-based complex event processing systems, to bridge the gap between different knowledge sources. For example, in IEC 62357 TC 57 Seamless Integration Architecture (SIA), an automation and power system management framework is proposed to define layers in smart grids as well as the

interfaces between layers.

The term *smart grid* refers to a modernization of the networks that connect electricity generation, transmission and distribution infrastructures, as well as the intelligent systems to control and optimize power flows. Accordingly, smart grids mainly consists of three layers which are physical layer, control layer and communication layer. The physical layer mainly supports power generation, power transmission and power distribution. The control layer is referred to the control system that is in charge with operating, controlling and monitoring core operations of power grid systems, e.g. Supervisory Control And Data Acquisition (SCADA) collects data from the physical layer and communicates with control system [3]. The communication network layer mostly includes control centers and field communication devices.

From the viewpoint of enterprise architectures, in particular ArchiMate [4], the components populate the technology layer. Besides the physical components such as generators, the firmware and their configurations of the cyber-physical components such as actuators and routers also belong to the technology level. The second ArchiMate enterprise architecture level defines applications and services. In the case of the smart grid, the services subsume power delivery, billing, maintenance planning, HR management and so forth, i.e. the application software running on SCADA and ERP servers. The third enterprise architecture level defines the business concepts, goals, functions and processes. From the viewpoint of the ELVIRA project that is driving this research, this level contains also the citizen goals, in particular the safe delivery of electric power to homes, to enterprises, and to other infrastructures.

The purpose of such models is to manage the IT-dependent assets of an enterprise. In this paper, we apply the same idea to manage the security of the IT-related assets. A particular property of power grids is the speed by which a malfunction of a component propagates in the grid and leads to immediate effects on connected components, in particular power consumers. We focus on the security aspects of cyber-physical components, i.e. components that are controlled by firmware and connected to SCADA systems. The physical power generation is considered to estimate the consequences of a cyber-attack. The research question of this paper is as follows:

How can a smart grid be represented in a repository such that it supports the vulnerability analysis against cyber attacks?

The repository provides static tools to assess the vulnerability of a smart grid, e.g. to evaluate whether the path between a SCADA program and the power grid components it controls are sufficiently protected by firewalls. The repository is able to store any number of smart grid models (test models, real models, variants). Further, the repository is able to export models in formats understood by network security simulators such as NS3 and by power grid simulators [5]. Such simulators provide dynamic analysis of smart grids. We also demand that the repository can represent events. Events are observations of state changes of the smart grid. In the next section, we discuss the taxonomy of smart grid components. Afterwards, the taxonomy is used as a domain-specific language to represent smart grids. Static analysis queries are used to pinpoint design weaknesses in the smart grid. Finally, we shortly elaborate how events can be added to the taxonomy and how this process allows to store attack patterns.

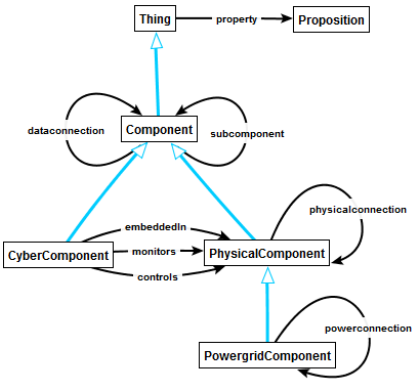

Figure 2. Top-level taxonomy for smart grids

II. Smart Grid Taxonomy

The description of the smart grid in the previous section is the basis to create a taxonomy of components. We use ConceptBase [6] to represent the taxonomy. This modeling tool has already been used in similar applications [7], [8]. A particular advantage of ConceptBase is its ability to represent both classes and objects in the same database. This allows us later to use the taxonomy as constructs of a domain-specific modeling language to represent sample smart grids to any degree of detail. It also allows to extend the taxonomy at any time, even when sample smart grids are already represented with the constructs of the taxonomy. Classes (taxonomy) and instances (sample models of smart grids) are stored in the same database.

Figure 2 displays the top-level constructs of the taxonomy starting with ”Thing”. It has a relation ”property” to the built-in class ”Proposition” of ConceptBase. This class is the most-general class of ConceptBase, subsuming any object. We use the ”property” relation to attach all property to smart grid components. The following properties are of particular interest:

- serialnumber: the external identifier of a smart grid component
- model: the given name of components model, e.g. the model name of a server computer
- version: the version of the model that was used to produce the component
- vendor: the vendor of the component

All such properties are optional. Note that some components like configuration files may not have serial numbers or vendors. Below ”Thing”, we define the sub-class ”Component”. Components subsume both physical components and cyber components. There are two relations for com-

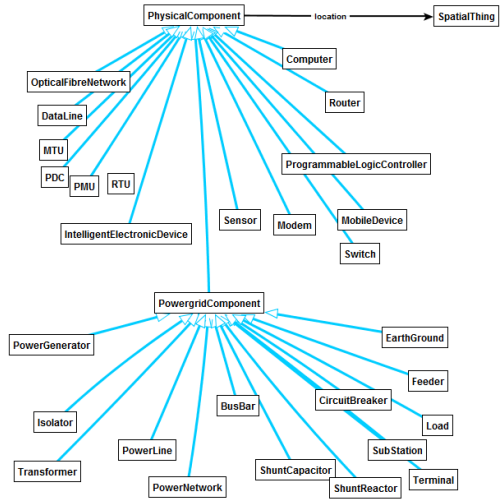

Figure 3. Physical and power grid components (excerpt)

ponents. First, components may be decomposed into subcomponents (product trees, bill of material). Second, there may be data connections between components. A data connection specifies that some data (measurements, control calls) may flow between two given components. This relation needs to defined at this level since sensors and actuators are directly connected to physical components. Cyber components are subsuming any type of computer-represented data or code. These components are embedded in physical components, e.g. a server computer or a router. Certain cyber components may monitor or control physical components.

A. *Components of the Power Grid*

Figure 3 displays the taxonomy of physical and powergrid components in our taxonomy. The entries were extracted from the literature study and by consulting with experts from the power sector.

Physical components have a location, which is typically a geographic area or just a single geographic point. Many such components are controlled and monitored by cyber components. Some physical components such as remote terminal units (RTU) are computerized, i.e. they have some firmware and they are connected to other computerized components, e.g. SCADA servers. We distinguish the hardware from the software running on such physical components. The software is discussed in the subsequent subsection. We list the computerized components all under physical components. Power grid components are special physical components that are used to generate, transmit, transform, and distribute electric power. The power grid components are subsuming those components that we identified in papers on analyzing power grids. The taxonomy is extensible at any time, even when it has been instantiated with power grid models. The components or subsystems in physical layer need to obey mainly functional requirements, for instance voltage tolerances, connected loads and lines, interconnections, and so forth. Power stations are physical areas that include among others power generators, transformers, busbars (connecting power grid components), circuit breakers (disconnect a power connection when needed). The power is transmitted to substations via power lines. A substation [9] has the main role to channel the power to other parts of the grid, in particular to distribution networks.

The components or subsystems for controlling the power grid need to obey mainly system requirements, for instance expandability, reliability, maintainability, data interconnections, etc. As the most common Industrial Control System (ICS), SCADA is an architecture of the control system that conducts process supervisory managements through data communications and operator interfaces with filed sensors and actuators. A SCADA system usually consists among others of RTU's, MTU's, HMI's, supervisory computers and communication infrastructure [10]. MTU's are used to forward the commands from SCADA system to RTU's. Although the communications between MTU's and RTU's are bidirectional, only MTU's can initiate the communication. RTU is an electronic device which monitors digital and analog parameters of objects in the physical layer, transmits telemetry data of the objects to the control layer, and controls the objects following the commands sent from the control layer. RTU's connect to sensors and actuators, and is networked to SCADA system [11]. PMU's are used to measure the electrical waves in real time and report to Phasor Data Concentrator (PDC). In power grids multiple remote measurements are conducted by using a synchronized-time source to support IEEE C37.118 standard for synchrophasor measurements for power systems. Human-machine interfaces (HMI) are used by humans to operate, monitor and control power components.

The SCADA components are connected via a communication network consisting of routers, switches, firewalls, data lines such as fiber optics cables. Control centers can be connected by multiple data lines to stations to collect data and manage the power production and distribution. A subset of the control layer components is listed in Figure 3. We distinguish cyber components (software, data) from the physical components that host them. The cyber components stand for actual computer readable code and data that is residing on some physical component. Firmware is executable code that runs on a device such as routers or RTU's. The firmware defines the behavior of the device, e.g. to which function calls it responds and how it interacts with connected components. A data stream is a stream of data records flowing in a data line that connects components. A configuration file is used to adapt a software to specific needs. Example of such a file could be the rules set for the firmware running on some firewall.

If the same operating system code is installed on two different components, then we regard them as two different cyber components (see Figure 4). They may share the model name and version number, but they are still different. This is important because one copy on one computer may be compromised while the copy on another computer could remain uncompromised. The same interpretation

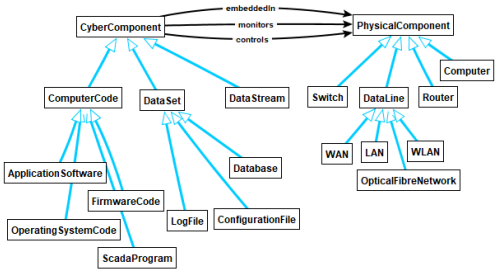

Figure 4. Cyber components (excerpt)

holds for data sets, configuration files, firmware, and even data stream. A data stream is assigned to data lines as their physical hosts. A data line is the counterpart of a power line in the power grid. It is physical by nature, even when using a wireless communication technology. A special case is the wide area network, which stands for a complex sub-system operated by internet-service providers.

We represent three types of relations between cyber components and physical components (Figure 4). The *embedding* specifies the fact that a cyber component is hosted on a physical component. The *monitoring* relation specified that a cyber component receives information from physical components, e.g. sensor readings from a transformer. The relation is logical, i.e. it may span over many physical components in between the host of the cyber component and the monitored component. For example, a SCADA program running on some server computer at control center 1 may monitor all transformers, busbars and generators of a given power station. Another SCADA program running on a different server computer may monitor the components of a distant sub-station. In both cases, there are many intermediate physical and cyber components that are required for the secure monitoring, such as RTU's, MTU's, switches, firewalls, and so forth. Finally, the *controls* relation specifies that a cyber component such as a SCADA program sends operation calls to physical components, such as actuators of power grid components.

III. Using the Taxonomy as Domain-Specific Language

The taxonomy discussed above provides an extensible list of components relevant to describe smart grids, i.e. power grids operated by complex IT systems. The concepts of the taxonomy form the classes to describe real or prototypical smart grids. ConceptBase allows to specify graphical symbols for such classes that apply to all instances of these classes. This results in a domain-specific graphical modeling languages for the components of a smart grid. Since our taxonomy covers both the cyber part and the power grid part, we can represent both views into a single model of a smart grid. ConceptBase also allows to extract views from such models, e.g. the representation of the power grid in a format readable by power grid simulation tools. Likewise, the cyber and network part can be extracted and passed to a network simulator.

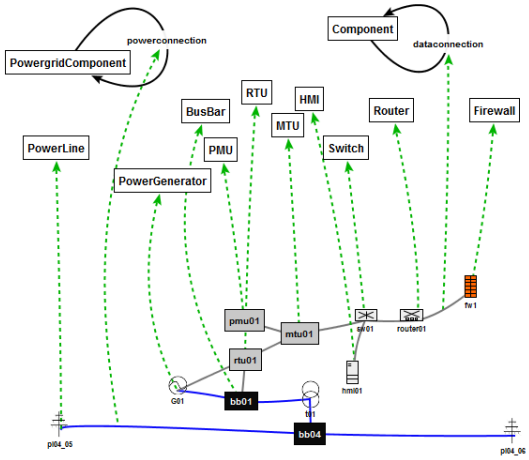

Figure 5. Power station represented as instance of the taxonomy

Figure 2 shows three basic connections between components. A power connection links two power grid components. The interpretation of a such a connection is that these two components can exchange electric power via this connection. A power connection itself is not a component. It is just stating that two components are connected. Power lines are used as intermediate components to bridge longer distances. Power connections are examples of physical connections. The third relation is a data connection. We define it for the class 'Component'. A data connection literally means that some data can flow between two components. The components may be physical entities such as computers, sensors, actuators, RTUs, transformers. Like power connections, data connections are not physical objects but express a logical relation. If larger distances have to be bridged, then one should use a data line components in-between, such as an optical fibre network.

Figure 5 shows the model of a power station represented as an instance of the taxonomy classes. The power generator G01 is connected via a power connection to busbar bb01, which itself connects to a step-up transformer t01. This transformer connects to a second busbar bb02, which has power connections to two power lines. The other components monitor and control the power components. The firewall fw1 has a data connection to a router which itself connects to mtu01 and then to pmu01 and rtu01. Note that the data connections can bridge between IT components such as the remote terminal unit rtu01 connects to busbar bb01. In reality, there may be a sensor component in-between, but its is left out of the model since the sensor unit is assumed to have no firmware that is subject to a cyber attack. The broken lines denote instantiation links to the taxonomy classes. The definitions of the smart grid components are made in a textual format:

```
G01 in PowerGenerator with
  nominalvoltage nc : "16.5KV" end
bb01 in BusBar with
  powerconnection pc1 : G01; pc2 : t01 end
router01 in Router with
  dataconnection dc1 : sw01 end
```

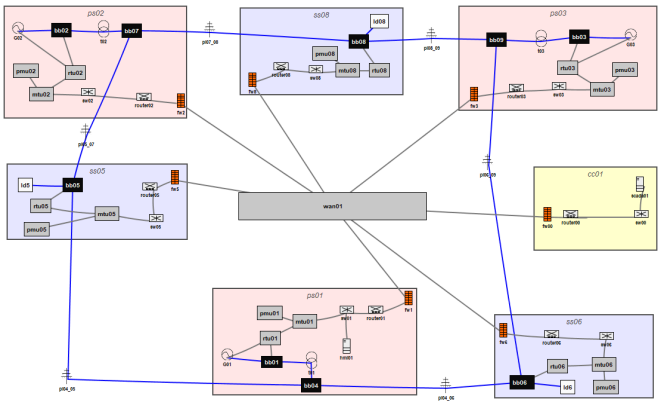

Figure 6. The IEEE 9-bus model with IT components; see fig. 5 for legend

ConceptBase allows to declare certain relations as symmetric via deductive rules. Thus, a data connection between two components is declared once and then interpreted as a symmetric relation. A similar rule exists for the power connections.

```
Component in Class isA Thing with
  attribute dataconnection : Component
  rule
    symrule_dc : $ forall c1,c2/Component
        :(c1 dataconnection c2):
        ==> (c2 dataconnection c1) $ end
```

Figure 6 shows the complete IEEE 9-bus [12] power grid extended by the IT components to control it. The model includes three power stations, three sub-stations, and one control center. The control center has a SCADA server scada01 the is connected via the wide-area network wan01 via firewalls to the sub-stations and power stations. Note that the power station model of Figure 5 appears as 'ps01' in the 9-bus model of Figure 6.

The wide-area network component wan01 stands for the Internet, i.e. for a whole communication infrastructure. Real power grids shall have a combination of Internet-based access, e.g. for remote sub-stations and dedicated data lines such as fibre optics' cables controlled by power-grid operators. Our taxonomy can cover such redundancy and repository services can be used to evaluate the level of redundancy via queries to the model. In the model, each of the stations employs a firewall to connect to the wide area network. The model shows which path an attacker can follow to reach a certain IT component, which then may affect the power generation, transmission and distribution. In the case of the attack on the Ukrainian power grid [13], the attack targeted the workstations of a control center and then operated the circuit breakers controlled by the SCADA system of that control center to shut down the power transmission over certain regions.

IV. Repository Services

The previous section shows how to use the smart grid taxonomy as a domain-specific language to represent integrated models of power grid and their IT components. The repository allows to manage any number of such integrated models, e.g. the IEEE 9-bus model. All these models are represented as instances of the taxonomy (see also figure 1. We use the module construct of ConceptBase to seperate the taxomomy from the models. The taxonomy is stored in one module and all models are stored as sub-modules of the taxonomy. Thus, the models share the definitions of the taxonomy but do not interfere with each other.

A. Reach of Cyber Components

Cyber components were defined as components embedded in physical components and controlling or monitoring other physical components, in particular power grid components such as circuit breakers. Of particular interest are SCADA programs because they monitor and control a large number of physical components. In the subsequent definition, the application software 'scadaprog1' is defined as being embedded in the SCADA server 'scada01' and controlling and/or monitoring a number of MTUs and RTUs:

```
scadaprog1 in ApplicationSoftware with
  embeddedIn host : scada01
  controls,monitors
   c1: mtu01; c2: mtu02; r1: rtu01; r2: rtu02
  controls
   c3: mtu03; r3: rtu03 end
```

In the example, the power grid components of power station 3 are only controlled but not monitored. The following query returns such power grid components:

```
Unmonitored in QueryClass
  isA PowergridComponent with constraint
  notmon : $ not exists sp/ApplicationSoftware
    sc/ScadaServer rt/RTU m/MTU stat/AnyStation
    (sp embeddedIn sc) and (sp monitors rt) and
    (sp monitors m) and (m dataconnection rt) and
    (rt dataconnection this) and
    (stat subcomponent this) and
    (stat subcomponent rt) and
    (stat subcomponent m) $ end
```

The answer to the query indicates, which components are not covered appropriately by a SCADA program. This may indicate an incompleteness in the model (component in monitored but the model is not reflecting this fact), or the designers of the SCADA system decided to leave certain components unmonitored. In large power grids, several control centers with SCADA servers are used to manage the grid. Thus, some components may be controlled by the SCADA system of one control center, and others utilize another control center. It may also be that two SCADA systems control and/or monitor the same physical component. This is for example the case when a power distributor operates a sub-station that is connected to the power grid operated by a transmission operator.

V. Discussion

An overview of testbeds for cyber-physical security testbeds for smart grids [14] lists the areas vulnerability, impact, impact, cyber-physical metrics, data and models, security validation, interoperability, forensics, and training. Our paper focuses on data and models for cyber-physical

systems. Hahn et al. [14] conclude that the research on data and models is particularly sparse on the physical aspect. Our taxonomy emphasizes the linkage between physical components and the cyber components that control the physical system.

A. Relation to Enterprise Architecture

The repository design was inspired by enterprise architecture systems such as ArchiMate [4]. Most of the smart grid components fall into the technology layer of ArchiMate. Some of the cyber components such as firmware, operating systems, database management systems etc. would also fall into this layer. Application software like SCADA programs belong to the ArchiMate application layer. The ArchiMate business layer is not yet used by our repository since we mostly focus on the infrastructure and not on business goals. The overarching business goal in our project is to secure the critical infrastructure of smart grids. The more smart functions are introduced to a smart grid, the more application programs become subject to protection. Hence, the business goals of providing more services to consumers may conflict with the goals to increase the security of the system.

A difference to enterprise architectures is that we represent both physical processes (power generation and distribution) and cyber processes (monitoring and controlling the physical processes). Enterprise architectures usually only focus on IT aspects. Another difference is the tight integration of the repository with analysis tools. The power grid topology of the integrated models of the smart grid can be exported to power simulation tools such as PySim. Such tools can compute the temporal changes of power distribution depending on the nomal power of generators and the changing power loads. Such analysis is useful to forecast the effect of putting certain components off service, e.g. by operating circuit breakers. Redundancy in the power grid may mitigate such events, but only simulation tools like PySim can forecast such effects in a reliable manner. Further, network simulators like NS3 may be used to analyse the effects of cyber attacks like denial of service attacks to the network. The corresponding network topology can be excerpted from the integrated models of the repository as well. The repository can support any number of models and variants in the repository. The goal of the ELVIRA project is to assess the vulnerability of smart grids against cyber attacks. Thus, we need to manage a large number of models to assess the effect of smart grid topologies and properties of their components on the vulnerability.

B. Expressiveness and Utility

The taxonomy and the smart grid models are represented in the Telos [15] language as implemented by ConceptBase. Their representation uses the so-called proposition (=object) data structure of Telos, which allows to handle links regular objects. ConceptBase represents all explicit information as objects, including classes, meta classes, their attributes/relations, specializations, and instantiation links. Deductive rules and queries are defined on top of this factual database (instances, classes, attributes, relations,...). As a consequence, the expressive power is as least equivalent to Datalog (with stratified negation). In this paper, the deductive capabilities are used to propagate properties such as the nominal voltage of power grid components. The query language is used to classify components based on the topology, for example the unmonitored power grid components in section IV-A. Any number of such classifying queries may be defined to assess the static properties of a smart grid. A particular strength of Datalog is its efficient implementation of recursive rules, such as the transitive closure of the data and power connection relations defined in the taxonomy.

Our goal was to develop an extensible taxonomy that can represent realistic smart grids containing a vast number of heterogeneous components. The IEEE 9-bus example was used as to validate this claim. We extended it by an IT layer to represent how SCADA servers monitor and control power grid components. A second goal was to provide a repository of smart grid models that can be analyzed by external tools such as simulators. We provide an XML-based export function that can generate the input format of various open-source simulators.

VI. Conclusions

We presented a taxonomy of smart grid components that can be used as a domain-specific language to represent smart grid models in a repository. The pupose of the repository is to serve as a database of smart grid models that are analyzed by external tools on their vulnerability. During the development, we continuously updated the taxonomy to cater for more component types. One advantage of ConceptBase tool is that such changes can be done even in the presence of example models. If new component types appear in the future, they can be added as subclasses of the existing component types. The stable part of the taxonomy are the notions of data connections between components, power connections between power grid components, and properties of any thing. While we target smart grids as application domain, the constructs of the taxonomy are also applicable to other cyber-physical infrastructures, e.g. industrial control systems. The arrival of the Internet of Things produces ever more devices that are connected to remote data collection and control servers. Another emerging application domain are smart homes, where the privacy and safety of citizens is at stake.

Source code of the implementation

The source code of the taxonomy and of some integrated models is provided via http://conceptbase.cc/elvirarep. Most of the figures in this paper are created with the graphical user interface of ConceptBase. The web page also provides information about the XML-based export function to analysis tools.

Acknowledgement

We thank all members of the ELVIRA project, in particular Birgitta Lindström, Sten Andler, Daniel Haglund, and Thomas Svensson for their input.

REFERENCES

[1] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*. IEEE, 2009, pp. 1–8.

[2] Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna, "Semantic information modeling for emerging applications in smart grid," in *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*. IEEE, 2012, pp. 775–782.

[3] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security - A survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017. [Online]. Available: https://doi.org/10.1109/JIOT.2017.2703172

[4] M. M. Lankhorst, *Enterprise Architecture at Work - Modelling, Communication and Analysis, 3rd Edition*, ser. The Enterprise Engineering Series. Springer, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-29651-2

[5] A. G. Wermann, M. C. Bortolozzo, E. G. da Silva, A. E. S. Filho, L. P. Gaspary, and M. P. Barcellos, "Astoria: A framework for attack simulation and evaluation in smart grids," *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 273–280, 2016.

[6] M. A. Jeusfeld, "Metamodeling and method engineering with ConceptBase," in *Metamodeling for Method Engineering*, M. A. Jeusfeld, M. Jarke, and J. Mylopoulos, Eds. MIT Press, 2009, pp. 89–168.

[7] P. Mason, "On traceability for safety critical systems engineering," in *12th Asia-Pacific Software Engineering Conference (APSEC 2005), 15-17 December 2005, Taipei, Taiwan*, 2005, pp. 272–282. [Online]. Available: https://doi.org/10.1109/APSEC.2005.85

[8] M. Vegetti, H. Leone, and G. Henning, "Pronto: An ontology for comprehensive and consistent representation of product information," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 8, pp. 1305 – 1327, 2011, semantic-based Information and Engineering Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197611000388

[9] J. D. McDonald, *Electric power substations engineering*. CRC press, 2016.

[10] S. A. Boyer, *SCADA supervisory control and data acquisition*. The Instrumentation, Systems and Automation Society, 2018.

[11] M. M. Ahmed and W. Soo, "Supervisory control and data acquisition system (scada) based customized remote terminal unit (rtu) for distribution automation system," in *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*. IEEE, 2008, pp. 1655–1660.

[12] S. Sharma, N. S. Velgapudi, and K. Pandey, "Performance analysis of ieee 9 bus system using tcsc," *Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, pp. 251–256, 2017.

[13] E-ISAC, "Analysis of the cyber attack on the ukrainian power grid - defense use case," Electricity Information Sharing and Analysis Center, Tech. Rep., 2016. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[14] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 847–855, 2013. [Online]. Available: https://doi.org/10.1109/TSG.2012.2226919

[15] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis, "Telos: Representing knowledge about information systems," *ACM Trans. Inf. Syst.*, vol. 8, no. 4, pp. 325–362, 1990. [Online]. Available: http://doi.acm.org/10.1145/102675.102676