Decision Support for Smart Grid: Using Reasoning to Contextualize Complex Decision Making

Marcello Balduccini*, Edward Griffor†, Michael Huth‡, Claire Vishik§, David Wollman†, Patrick Kamongi†

*Saint Joseph's University, Philadelphia, PA, USA; marcello.balduccini@sju.edu

†US National Institute of Standards and Technology, Gaithersburg, MD, USA; {erg1,patrick.kamongi,wollman}@nist.gov

‡Imperial College London, London, UK; m.huth@imperial.ac.uk

§Intel Corporation, Austin, TX, USA; claire.vishik@intel.com

***Abstract*—The smart grid is a complex cyber-physical system (CPS) that poses challenges related to scale, integration, interoperability, processes, governance, and human elements. The US National Institute of Standards and Technology (NIST) and its government, university and industry collaborators, developed an approach, called CPS Framework, to reasoning about CPS across multiple levels of concern and competency, including trustworthiness, privacy, reliability, and regulatory. The approach uses ontology and reasoning techniques to achieve a greater understanding of the interdependencies among the elements of the CPS Framework model applied to use cases. This paper demonstrates that the approach extends naturally to automated and manual decision-making for smart grids: we apply it to smart grid use cases, and illustrate how it can be used to analyze grid topologies and address concerns about the smart grid. Smart grid stakeholders, whose decision making may be assisted by this approach, include planners, designers and operators.**

I. Introduction

The *smart grid* is a system of systems, including instances of Cyber-Physical Systems (CPS) and Internet of Things (IoT), that exhibits both scale and horizontal integration. *Scale* does not merely refer to the number of systems in a smart grid. but relates to their dynamic and coordinated function, and the interactions needed to make an electric grid *smart*. Smart grid exhibits *complexity* in technological and functional diversity, as well as diversity of ownership of its components.

Horizontal integration – e.g. between smart grid, smart street lights, smart homes, and electric transportation - has considerable complexity, and no sole horizontal technology platform has been shown to span all aspects of a smart grid. Moreover, a single platform could form a single point of failure or pose additional cyber risks.

The analysis of complex systems is improved by using specialized models or frameworks. Models applicable to smart grid exist, e.g., the NIST CPS Framework.

Smart grid can benefit from knowledge representation and reasoning tools. The use of ontologically inspired modeling in computer science is not new. As Smith and Welty [1] point out, this approach has been used extensively in information and computer science, including database development or domain modeling in software engineering.

Existing frameworks can speed up ontology development, thus creating premises for reasoning and decision support applications. In this case, the authors had the advantage to rely on the NIST CPS Framework, created by a NIST facilitated Public Working Group. A key outcome of that work is the CPS Framework (Release 1.0, published as three separate NIST Special Publications [2], [3], [4]), which proposes a means of describing three *facets* during the life of a CPS: conceptualization, realization, and assurance of CPS; and to facilitate these descriptions through analytical lenses, called *aspects*, which group common concerns addressed by the builders and operators of the CPS. In the framework, the aspect named *Trustworthiness* describes multiple related *concerns* that deal specifically with the avoidance of harm in privacy, security, safety, resilience, and reliability. The framework is extensible and supported with additional models, e.g. a UML model of concerns, aspects, all three facets, and the interdependencies across the CPS lifecycle.

The CPS Framework describes the activities and artifacts of CPS development in a precise way and enables concerns that motivate important requirements to be considered in conceptualizing, realizing (including operating), and assuring CPS. However, the CPS Framework does not, by itself, include the ability to reason about CPS. In this short paper, we propose to use ontology based reasoning to realize such capabilities for smart grid use cases with a focus on trustworthiness. The paper extends our prior work on reasoning for the CPS Framework [5]. In this paper, we demonstrate that the approach extends naturally to automated and manual decision-making for smart

M. Balduccini was partly supported by NIST grant 70NANB17H260 and 70NANB18H257.

grids. We apply it to smart grid use cases, and illustrate how it can be used to analyze grid topologies and address concerns about the smart grid. Additionally, we substantially expand the the representation of properties, concerns, and their links, and introduce measures of the optimality of solutions.

The model contains sufficient complexity to demonstrate the capabilities of the approach and its applicability to smart grid infrastructures. The case study of this paper includes, e.g., considerations such as transduction (where a CPS produces a physical signal that interacts with the environment) and influence (where a CPS produces or receives a physical signal causing a state change of another CPS).

II. Related Work: Ontology-Based Reasoning

Recognizing the complexity of a smart grid, some researchers have turned to efforts that include ontologies and reasoning. Among notable research papers published recently, we can mention an ontology for energy management in smart grid [6] and an ontology focusing on smart grid knowledge exploitation [7], [8]. Ontologies for security and trustworthiness have also been pursued by researchers in recent years. Examples include work on ontologies for certification and testing [9], and work on ontology integration in security [10].

However, there has been no work to date on ontology-based reasoning for the trustworthiness of smart grid.

III. A CPS Reference Framework

A CPS often delivers complex functions that are ultimately implemented in diverse inter-operating systems and devices. Interactions can occur through the exchange of information or the exchange of matter/energy. The former are *logical* interactions and the latter *physical* interactions.

The functional decomposition of a CPS breaks it down, from a name and brief description of what the system is or does – the *Business Case* – through the set of scenarios or step-by-step descriptions of ways of using the system and the functions that realize those steps – the *Use Case* – the actors/subsystems and interactions – the *Allocation of Function* – and to the allocation of given subsystem functions to physical or logical implementation – *Physical-Logical Allocation*.

Concerns about CPS/IoT are represented in a forest, where trees and branching corresponds to the *decomposition of concerns*; see Fig. 1 for a view of the Trustworthiness *concern tree*. We refer to this structure as the *concern forest* of the CPS Framework. The concerns at the roots of this structure, the highest level concerns, are called *aspects*; there are nine aspects, one of which being *Trustworthiness*.

A concern about a given system reflects a dimension of issues to be addressed in the realization of a CPS. A *property* is a requirement or statement that addresses a concern. This method or practice is applied to each function in the functional decomposition of the system. A concern can be uniquely identified with a branch in the concern forest, and can be represented as consisting of a root followed by a (possibly empty) sequence of concern names in the branch, separated by dots. In the *Trustworthiness* aspect, e.g., we have the concern $TW.Security.Cybersecurity.Confidentiality$, which may be abbreviated as, e.g., $Conf'd$. A sample property, meant to address this concern about data exchanged between components of a system, is “all communications shall be encrypted via AES.”

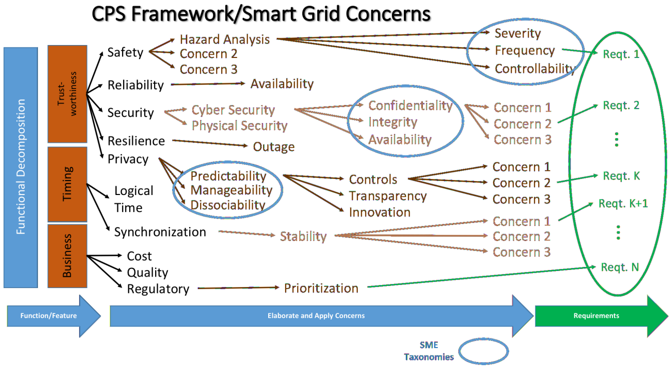

Fig. 1: Branching for smart grid concerns

The Framework provides guidance on how to develop an *Assurance Case for each concern applied to the CPS*, comprised of: properties of the CPS and the concerns that resulted in their addition to the model of the CPS; *argumentation* or criteria for concluding that a property has been established of the CPS; *evidence* information, accessible to stakeholders, that the criteria used in this argumentation are indeed met; and *uncertainty* associated with the evidence that the criteria are met. The framework has been applied to complex environments including CPS, such as smart grid, and provides a structured way to analyze complex environments. For example, the Framework provides the ability to develop new management tools, such as those based on UML/XML modeling approaches, that are essential to understanding critical performances of CPS incrementally, in CPS development, deployment, adoption and operation.

IV. A CPS Framework Ontology

In order to develop reasoning capabilities for the CPS framework, we developed an ontology of the CPS Framework [11]. An ontology is a formal, logic-based representation of knowledge supporting reasoning by means of logical inference. In this paper, we adopt a broad view of this term: by “ontology” we mean a collection of statements in a logical language that represent a given domain in terms of *classes* (i.e., sets) of objects, *individuals* (i.e., specific objects), relationships between objects and/or classes, and logical statements concerning these relationships. For example, an ontology focusing on the trustworthiness of CPS may define the high-level concept of “Concern” with its refinement of “Aspect.” All of these are formalized as classes and, for Aspect, subclasses. Specific concerns are represented as individuals: TW as an individual of class Aspect, $Security$ and $Cybersecurity$ as individuals of class Concern. Also, a relation “has-subconcern” associates a concern with its sub-concerns. Thus, Aspect “has-subconcern” $Security$, which in turn “has-subconcern”

$Cybersecurity$. By introducing a property "satisfied," one can also indicate which concerns are satisfied.

Inference can then be applied to propagate "satisfied" and other relevant properties and relations through this ontology. For example, given a concern that is not "satisfied," one can leverage relation "has-subconcern" to identify other concerns that are not satisfied because of it, either directly or indirectly.

In practice, it is often convenient to distinguish between the factual part, Ω, of an ontology and its *axioms*, Λ. The former, from now on simply called "ontology," encodes the factual information, e.g., TW "has-subconcern" $Security$. The latter expresses deeper, often causal, links between relations, e.g. that a concern is not satisfied if any of its sub-concerns is not satisfied.

V. Applying CPS Framework to the Smart Grid

Our approach to reasoning leverages a logic-based representation of a system of interest and applies inference to draw new and useful conclusions in a rigorous way. It is agnostic to specific choices of logical language and inference mechanism. It assumes the existence of axioms in the selected logical language, which formalize the queries one is interested in answering, the type of reasoning to be carried out, and any contextual information. Conclusions are drawn from an ontology Ω and a set of axioms Λ by means of a logical inference procedure, denoted by symbol $\vdash$. If Δ follows from Ω and Λ, we write $\Omega \cup \Lambda \vdash \Delta$, where $\cup$ denotes set union.

For example, in the context of cybersecurity, the language of *propositional logic* can be used to represent (a) that a cyberattack occurred (statement p) and (b) expert knowledge that, when that cyberattack occurs, a certain system becomes inoperative (statement $p \supset q$, read "p implies q," where q states that the system is inoperative). The logical inference $\{p\} \cup \{p \supset q\} \vdash q$ allows one to draw the conclusion that q holds, i.e. that, as a result of the cyberattack, the system is expected to be inoperative.

A. Formalizing the smart grid

For the purpose of illustrating the importance of reasoning for decision support on smart grid, we divide the *reasoning space* up into *layers* and illustrate how reasoning can bring benefits, both within and across, these layers:[1]

- Component (Generation, transmission, distribution, DER [distributed energy resources], customer premises)
- Communication (Protocols used to deliver, share, and communicate data and information)
- Information (Models for data and information transmitted during smart grid operations)
- Function (Use cases and functions for smart grid operations)
- Business/Environment (Legal, regulatory, policy, economic framework)

For the sake of illustration of how systems forming smart grid are analyzed using reasoning, we consider a use case centered around the self-healing function for resilience and safety, and consider as well potential concerns about privacy related to this function.

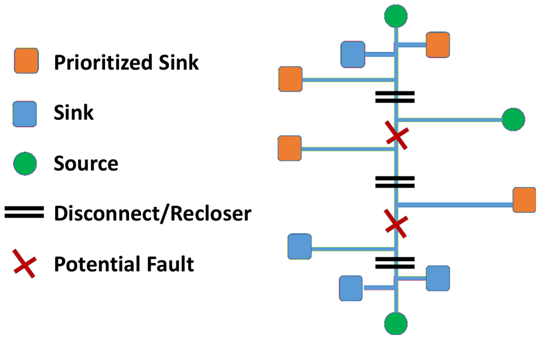

Fig. 2: Diagram of smart grid scenario

The scenario for this use case, depicted in Fig. 2, considers fault(s) on a distribution line affecting consumers including several that are prioritized. The smart grid may be designed to enable intelligent, communicating reclosers to work together to isolate the fault and minimize the extent of the power outage and restore power to as many customers as practical, while respecting priorities, e.g. critical infrastructure or critical care patients for service and restoration. Incumbent on the designers and implementers of the smart grid is to evaluate and enable decision making that results in the lesser over the greater risk of harm, including safety of customers and utility workers. The recovery action is to re-route power to achieve this end, using *loops* in the smart grid topology. This type of situation can be complex with multiple sources of power, possible disconnects and prioritized customers. Fig. 2 is abstracted from existing reference grid Bus systems (e.g. IEEE Bus System 57 and IEEE 300).

In this example, the system includes a situational awareness and decision module (SADM), which controls the system's configuration and processes. This use case is chosen because it encompasses major component types of a CPS, raises key trustworthiness concerns and lends itself to various non-trivial investigations. It also denotes a typical activity in a smart grid, in this case, ensuring resilience and safety. Through this use case, we highlight the interplay among trustworthiness concerns, as well as their ramifications on other CPS aspects.

For the sake of simplicity, we assume that the disconnects are capable of two modes (open-disconnected or closed-operational), and the system can access any available configuration of connected sources and sinks with some sinks (customers) identified as priorities. It is assumed that the topology model for the smart grid infrastructure exists (comprising component, communication, information, function and business/environment layers) and includes state information about disconnects (communication layer), prioritizations (business/environment layer) and other relevant operational status parameters, such as line capacity, temperature, etc. A problem/issue is additional information related to faults and their approximate location on the grid topology.

In our approach, the formalization of a CPS is organized

[1] https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf

along multiple levels: (L1) aspects and concerns; (L2) properties; (L3) CPS configuration; (L4) actions; (L5) constraints, dependencies and trade-offs; and (L6) satisfaction axioms. Level L1 and L6 form the *CPS-independent specification*, since aspects and concerns are independent of the specific CPS being modeled. Levels L2-L5 comprise the *CPS-dependent specification*, as the information included in them depends on the CPS being modeled. From another perspective, levels L1 and L2 formalize the concepts from the definition of the CPS Framework. Levels L3-L5 extend the framework to provide details needed for reasoning about the behavior of a CPS of interest. Level L6 provides the semantics of the formalization.

Formalization of aspects and concerns. The formalization of aspects and concerns is shared by all CPS. The nodes of a concern tree are represented by individuals of class *Concern*. The root nodes of the concern trees are a particular kind of concern, and so they are placed in a class (*Aspect*) that is a subclass of *Concern*. Following the definition of the CPS Framework, class *Aspect* includes individuals *Trustworthiness*, *Timing* and *Functional* for the corresponding aspects, while class *Concern* includes individuals *Security*, *Cybersecurity*, *Functionality*, etc.

Edges linking aspects and concerns are represented by the relation subConc, which is a representation of the notion of sub-concern. Thus, an edge from a concern x to a concern y is formalized by a statement $\mathsf{subConc}(x, y)$. Statement $\mathsf{subConc}(TW, Security)$, e.g., formalizes that the Security concern is a direct sub-concern of the *Trustworthiness* aspect. Concerns $Cybersecurity$ and $Conf'd$ are linked similarly.

Formalization of a CPS instance. The specific configuration of a CPS instance is formalized by suitable classes, subclasses, and individuals. For example, the nodes of a smart grid can be described by means of a class $node$, with subclasses $sink$ and $source$. Specific sinks and sources in the grid are represented by individuals of the corresponding classes. To enable writing logical formulas that mention individuals and the classes they belong to, we introduce proposition $is\_a(i, c)$, which holds when individual i belongs to class c, possibly via intermediate subclasses[2].

Formalization of properties. Properties of a CPS are specified by logical formulas over propositions. A property "faulty element[3] e_1 shall be de-energized" could be formalized as $faulty(e_1) \supset \neg energized(e_1)$. As shown in this example, properties and configurations can be negated by prefixing them by symbol $\neg$. For ease of representation, we allow formulas to contain variables (denoted by uppercase initial). As in first-order logic, a formula containing variables can intuitively be viewed for an abbreviation of the set of formulas obtained by replacing its variables by all possible constants. Thus, a requirement "all faulty elements shall be de-energized" can be expressed by the formula $is\_a(E, element) \wedge faulty(E) \supset \neg energized(E)$.

Advanced aspects of the model are formalized by means of the specification language shown in Table I. Specifically, an edge that links a property with a concern it addresses is represented by a property-concern link statement. For example, the fact that the above property addresses the *Safety* concern is formalized by a statement:

$$\begin{aligned} &is\_a(E, element) \ \wedge \ faulty(E) \ \supset \\ &\quad \neg energized(E) \text{ addresses } TW.Safety \ \ [\mathbf{10000}] \end{aligned} \tag{1}$$

A weight can be associated with property-concern links to indicate the "cost" of failing to address the concern through the property. This allows to reason about suboptimal situations in which concerns may be unsatisfied. Intuitively, the higher the weight, the more important the corresponding property is. For instance, a situation in which two faulty elements are energized will have a weight of 20000. Depending on the weights associated with other property-concern links, this may indicate a highly undesirable situation. A decision-support system evaluating potential solutions to a problem in a smart grid will then avoid those with such high weights.

Formalization of actions. We use the term "action" to denote both those actions that are within the control of an agent, e.g., actions an operator may take, and those actions that occur spontaneously, such as a disconnect that automatically opens when it senses a fault nearby. The formalization includes a suitable class *Action* and individuals for the actions of interest. In the case of a smart grid, one might introduce actions $open(d)$ and $close(d)$ to formalize the actions of opening and closing a disconnect d.

Formalization of observations and action occurrences. The observation that a proposition π holds in the current state of a CPS is captured by a statement of the form $obs(\pi, true)$ (resp., $obs(\pi, false)$ if the proposition is observed to be false). The hypothesized occurrence of an action a at some point s in the evolution of the CPS is represented by $occurs(a, s)$. (The notion of step in the evolution of the CPS is discussed later.)

Formalization of proposition dependencies, defaults and triggers. The remaining statements from Table I are inspired by research on action language $\mathcal{AL}$ [12] and enable the specification of further details about the model.

Statement type	**Syntax**
Property-concern link	• Γ addresses γ
Proposition dependency	• π if Γ
Default proposition value	• π defaults $true$ • π defaults $false$
Effects of actions	• a causes π if Γ
Triggered actions	• Γ triggers a

TABLE I: Specification of properties, dependencies, trade-offs; Γ, π range over (sets of) propositions, a over actions and γ over concerns

A *proposition dependency statement* states that, whenever all propositions in Γ hold, π also holds. For instance, the statement $energized(SRC)$ if $is\_a(SRC, source) \wedge active(SRC)$ captures the intuition that a source that produces power is energized. The specification of *default proposition values* is useful when information about the state of the CPS is incomplete. For instance, one can use the following statement

[2] Technically speaking, this is achieved via transitive closure.

[3] See below for details on the notion of element in the formalization.

to specify that disconnects should be assumed to be in working order (i.e., not stuck) in the absence of contrary evidence: $stuck(D)$ defaults $false$. The next type of statement describes the *effects of actions*, such as opening a disconnect that is working properly:

$$\begin{aligned} &open(D) \text{ causes } \neg closed(D) \text{ if} \\ &\qquad\qquad is\_a(D, disconnect) \;\wedge\; \neg stuck(D) \end{aligned}$$

The last type of statement from Table I describes the spontaneous triggering of actions when certain conditions are satisfied. Consider the case of a disconnect that is capable of automatically opening if it senses that a nearby node has become faulty. This can be formalized by the trigger:

$$\begin{aligned} &is\_a(D, auto\_disconnect) \;\wedge\; nearby(D, N) \;\wedge \\ &\qquad\qquad faulty(N) \text{ triggers } open(D) \end{aligned}$$

Axioms. Recall that our approach reduces the task of answering a query of interest to that of finding one or more answers, Δ, such that $\Omega \cup \Lambda \vdash \Delta$ holds, where the ontology Ω and any supporting axioms Λ are expressed in a logical language for the reasoner of choice. Set Λ contains the encoding of all statements introduced above together with statements formalizing their semantics. One such statement captures the intuition that:

> *A concern is satisfied when all properties addressing it and all sub-concerns are satisfied.* (2)

The content of Λ depends on the logical language of choice. Given a suitable form of implication $\leftarrow$, e.g., a default proposition value statement can be translated to $holds(\pi, S) \leftarrow holds(\pi_1, S), holds(\pi_2, S), \ldots, holds(\pi_k, S)$, where $\Gamma = \{\pi_1, \ldots, \pi_k\}$. A thorough discussion on this topic is beyond the scope of this paper and thus here we rely on the statements' informal semantics in order to draw conclusions. Using this approach, notice that axiom (2) is responsible for recursively propagating the satisfaction of properties and concerns, or lack thereof, up the relevant concern tree. Thus, if a faulty element is energized, (1) makes it possible to conclude that the *Safety* concern is not satisfied, and (2) concludes that the *Trustworthiness* aspect is not satisfied.

B. Application to Decision-Support

In this section, we illustrate how our approach can be applied to the development of decision-support systems for the smart grid. For use in decision-support, set Λ is augmented with a reasoning module μ formalizing the reasoning task of interest. We will see an instance of that later in this section. To a large extent, however, reasoning modules can be written once and for all and are, to a large extent, independent of the problem instance and, in fact, even of the application domain.

Recall that our focus in this paper is on decision-making techniques capable of spanning multiple levels of abstraction and of concerns. Thus, in the discussion that follows we abstract away from the fine-grained details of the components of the power grid and rather focus on a high-level description of scenarios such as the one from Figure 2.

Let us model the grid as a collection of *elements*, further distinguished in *nodes* and *links* – the latter corresponding to conductive elements that connect nodes. Nodes are divided in *sources* that output power, *sinks* that consume it[4], and *junctions,* where the links corresponding to multiple branches of a grid are connected. Sinks are further divided in *prioritized sinks*, which must be given particular attention, such as customers with life-safety energy requirements, and *non-prioritized* ones. Proposition $connected(N_1, N_2, L)$ states that the corresponding nodes are connected by link L. *Disconnects* can be used to control power flow through links. This is formalized by a proposition $controls(D, L)$. Further propositions are shown in Table II. The table also lists the available actions.

Proposition	**Meaning**
$closed(D)$	D is closed
$faulty(E)$	E is faulty
$active(SRC)$	SRC is active, i.e. producing power
$energized(E)$	E is energized
Action	**Meaning**
$open(D); close(D)$	open/close D
$enable(SRC); disable(SRC)$	enable/disable SRC

TABLE II: Additional propositions and actions; E ranges over elements; D over disconnects; SRC over sources

Arguably, one of the major concerns when faults occur in a grid is safety. One property that addresses safety is that "all faulty elements shall be de-energized". This information can be formalized by means of statement (1) shown earlier. Furthermore, from a regulatory perspective, one will want to minimize the number of impacted customers. More precisely, the *Business.Regulatory* concern might be addressed as follows:

$$\begin{aligned} &is\_a(S, prioritized\_sink) \supset energized(S) \\ &\qquad\qquad \text{addresses } Business.Regulatory \;\; [\mathbf{2}] \\ &is\_a(S, non\_prioritized\_sink) \supset energized(S) \\ &\qquad\qquad \text{addresses } Business.Regulatory \;\; [\mathbf{1}] \end{aligned} \tag{3}$$

The statements embody the connection between the regulatory concern and the property that "all sinks shall be energized." The (notional) weights indicate that serving prioritized sinks has greater relative importance. In case it is impossible to energize all sinks, solutions should privilege prioritized sinks.

We will illustrate our approach by demonstrating how it can be used to answer important questions about the example scenario. Most of the underlying reasoning tasks are centered on determining whether a certain expression χ is true in a state s of the CPS[5], captured by an expression of form $holds(\chi, s)$ where χ can be an arbitrary proposition or special expression $sat(\gamma)$ – indicating satisfaction of a concern γ.

Concern tree. Given information about the state of the smart grid, the first task of interest is checking which concerns are satisfied. State information is given by $obs(\cdot, \cdot)$ statements. For instance, the fault at the top of Figure 2 might be described by $obs(faulty(link_9), true)$. Let us suppose that no sources

[4] Sinks capable of outputting power can also be incorporated in the model.

[5] At this level of abstraction, the evolution of the state is characterized in terms of discrete time steps.

are active in the smart grid. By inspecting statements (1) and (3), it is not difficult to see that the *Safety* concern is satisfied, while the *Regulatory* concern is not. This conclusion can be reached formally by checking, e.g., whether $\Omega \cup \Lambda \vdash holds(sat(TW.Safety), 0)$. The same method also allows one to derive that *Trustworthiness* is satisfied and *Business* is not.

What-if. Suppose an operator (human or automated) faced with the above scenario would like to evaluate ways to bring power to the sinks. This decision-support task can be tackled by means of *What-if* reasoning, which studies how the CPS might be affected by potential actions. A query "is χ satisfied at step s?", is answered by checking whether $\Omega \cup \Lambda \vdash holds(\chi, s)$. To check the effect on *Safety* of activating the source at the top of Figure 2, one can expand Λ to include $occurs(enable(src_1), 0)$ and check $\Omega \cup \Lambda \vdash holds(sat(TW.Safety), 1)$. It is not difficult to see that the expression does not hold, because both faulty nodes are now energized. (Recall that disconnects are assumed closed for illustration purposes.) This provides the operator with useful information for evaluating the proposed course of action.

Mitigation. While the what-if reasoning task can help an operator evaluate potential solutions, our approach can also be used to automatically or semi-automatically compute solutions to smart grid problems. This is achieved by posing the query "which course of action can lead to the satisfaction of concern γ?" Let us assume that the underlying logical language supports disjunctive statements of the form $p \vee q$ (true if at least one of p, q is true) and let us expand Λ by $occurs(a, s^{\#}) \vee \neg occurs(a, s^{\#})$ for every action a and step s within some time horizon $s^{\top}$ of interest. Intuitively, this statement allows the reasoner to consider various possible courses of actions. If some action occurrences are included in Λ, then the task is semi-automatic, and the decision-support system looks for completions of the given course of action. The question is answered by computing actions $a_1, \ldots, a_k$ such that $\Omega \cup \Lambda \vdash \{holds(sat(\gamma), s^{\top}), occurs(a_1, s_1), \ldots, occurs(a_k, s_k)\}$. In case multiple solutions are found, their weights can be used to rank them based on desirability. In our scenario, suppose the operator wants to find a course of action that restores the *Safety* concern after activating src_1. One possible answer to the corresponding query is that the disconnect at the top of the diagram in Figure 2, say d_1, should be opened. Note that this solution has a weight of 7 due to the sinks (prioritized and not) that are left without power, causing the *Regulatory* concern to be unsatisfied. Another possible answer additionally prescribes the activation of the source at the bottom of the diagram and the opening of the nearby disconnect. The *Safety* concern is still satisfied, but this solution is more desirable, as it has a weight of 5 (cfr. (3)). Also note that the course of action in which all sources are enabled and all disconnects opened has a greater weight, since the fault at the top becomes energized.

Note that variations of these reasoning tasks are possible, such as finding courses of action satisfying all concerns, as well as more sophisticated reasoning tasks, e.g. decision-support for diagnosis. A more thorough discussion will be the subject of a longer version of this paper.

VI. Conclusion

In this paper, we use the CPS Framework and an abstract model for smart grid scenarios related to resilience to illustrate ontology-based decision support for smart grid. We demonstrate the ability to gain additional insights into smart grid use cases through reasoning – particularly critical since recent developments, e.g. micro grids and blockchain, are set to add complexity to smart grid management. Although the use case illustrates a simple activity associated with smart grid operations, it highlights the ability to perform sophisticated analysis, e.g. on consequences of proposed fault mitigations. We believe the model contains sufficient complexity to demonstrate the capabilities of the approach and its applicability to smart grid infrastructures. Aspects of smart grids and their management, involving design and operations for regulation and technology integration may specifically benefit from our approach. Our work illustrates how an ontology-based methodology, assisted by logic-based reasoning, can aid engineers, operators, leaders in identifying and resolving issues in design, operation, and assurance of the CPS that support smart grid infrastructures.

References

[1] B. Smith and C. Welty, "Ontology: Towards a New Synthesis," *Formal Ontology in Information Systems*, vol. 10, no. 3, pp. iii–x, 2001.

[2] E. Griffor, C. Greer, D. Wollman, and M. Burns, "Framework for Cyber-Physical Systems: Volume 1, Overview," National Institute of Standards and Technology, Tech. Rep. NIST-SP-1500-201, Jun 2017. [Online]. Available: https://www.nist.gov/publications/framework-cyber-physical-systems-volume-1-overview

[3] ——, "Framework for Cyber-Physical Systems: Volume 2, Working Group Reports," National Institute of Standards and Technology, Tech. Rep. NIST-SP-1500-202, Jun 2017. [Online]. Available: https://www.nist.gov/publications/framework-cyber-physical-systems-volume-2-working-group-reports

[4] D. Wollman, M. Weiss, Y. Li-Baboud, E. Griffor, and M. Burns, "Framework for Cyber-Physical Systems: Volume 3, Timing Annex," National Institute of Standards and Technology, Tech. Rep. NIST-SP-1500-203, Sep 2017. [Online]. Available: https://www.nist.gov/publications/framework-cyber-physical-systems-volume-3-timing-annex

[5] M. Balduccini, E. Griffor, M. Huth, C. Vishik, M. Burns, and D. Wollman, "Reasoning about Smart City," in *4th IEEE International Workshop on Sensors and Smart Cities*, 2018.

[6] P. Brizzi, D. Bonino, A. Musetti, A. K. E. Patti, and M. Axling, "Towards an ontology driven approach for systems interoperability and energy management in the smart city," in *Computer and Energy Science (SpliTech), Int'l Multidisc. Conf.* IEEE, July 2016, pp. 1–7.

[7] P. Bellini, I. Bruno, A. Cavaliere, D. Cenni, M. DiClaudio, G. Martelli, N. Rauch, and et al., "Km4City: Smart City ontology and tools for city knowledge exploitation," in *European Data Forum*, 2015.

[8] N. Komninos, C. Bratsas, C. Kakderi, and P. Tsarchopoulos, "Smart city ontologies: Improving the effectiveness of smart city application," *Journal of Smart Cities*, pp. 31–46, 2016.

[9] J. S. Luciano, D. S. Sayogo, W. Ran, N. Depaula, H. Jarman, L. Luna-Reyes, D. F. Andersen, and et al., *Private Data and Public Value*. Springer, 2016, ch. Using Ontologies to Develop and Test a Certification and Inspection Data Infrastructure Building Block, pp. 89–107.

[10] C. Porcel, C. Martinez-Cruz, J. Bernab-Moreno, A. Tejeda-Lorente, and E. Herrera-Viedma, "Integrating ontologies and fuzzy logic to represent user-trustworthiness in recommender systems," *Procedia Computer Science*, vol. 55, pp. 603–612, 2015.

[11] M. Balduccini, E. Griffor, M. Huth, C. Vishik, M. Burns, and D. A. Wollman, "Ontology-based reasoning about the trustworthiness of cyber-physical systems," *CoRR*, vol. abs/1803.07438, 2018.

[12] M. Gelfond and V. Lifschitz, "Action Languages," *Electronic Transactions on AI*, vol. 3, no. 16, 1998.