Reliability Engineering and System Safety 165 (2017) 89–101

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Hybrid systems modeling for critical infrastructures interdependency analysis[☆]

Constantinos Heracleous, Panayiotis Kolios, Christos G. Panayiotou[*], Georgios Ellinas, Marios M. Polycarpou

KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus

ARTICLE INFO

Keywords:
Critical infrastructure systems (CISs)
Interdependencies analysis
Hybrid systems
Open hybrid automata
Cascading failures

ABSTRACT

Critical infrastructure systems (CISs) are large scale and complex systems, across which many interdependencies exist. As a result, several modeling and simulation approaches are being employed to study the concurrent operation of multiple CISs and their interdependencies. Complementary to existing literature, this work develops and implements a modeling and simulation framework based on open hybrid automata to analyze CISs interdependencies. With the proposed approach, it is possible to develop accurate models of infrastructure components, and interlink them together based on their dependencies; in effect creating larger and more complex models that incorporate interdependencies. By implementing specific setups using varying operating conditions, one can study the cascading effects of interdependencies, perform a detailed vulnerability assessment and conduct an extensive planning exercise. To demonstrate the applicability of the proposed framework, a setup with three different types of CISs (i.e., power, telecom and water) components is investigated. Extensive simulation results are used to provide insights on the cascading effects, vulnerabilities and maintenance planning strategies.

1. Introduction

Critical infrastructure systems (CISs), such as power distribution systems, telecommunications networks, and water distribution networks, provide the necessary services that are vital to the security and well-being of the society. Disruption, damage or complete destruction to these infrastructures, due to natural disasters, accidents or malicious attacks, can have significant negative consequences and thus actions, to improve their protection and reliability, are of paramount importance [1].

Although each CIS is usually treated as an individual system, all CISs are highly interconnected with various interdependencies among them. For example, communication systems need a steady supply of electricity to maintain a good quality of service (QoS), while electric systems need reliable communications to maintain an accurate system state estimation. These bidirectional relationships between infrastructures enhance their overall performance, but at the same time increase their complexity and vulnerability [2,3].

Interdependencies are often unnoticeable when CISs maintain their normal operations, however, they can become critical during failures (e.g., due to operation errors, aging, poor maintenance etc.), deliberate attacks and natural disasters [4]. Moreover, cascading interdependencies can increase the scale of destruction in multiple CISs. This was observed in a number of events worldwide, such as the 2001 World Trade Center Attack [5], the 2005 Hurricane Katrina [6], the 2011 Fukushima Daiichi nuclear disaster [7], and several others [8,3]. Interdependencies, based on their characteristics and effects on infrastructures, are classified into the following four principal types [2]: (a) *physical*, if the operations of one infrastructure depends on the physical output(s) of the other and vice versa, (b) *cyber*, if there is information/signal transmission between different infrastructures, (c) *geographic*, if components of different infrastructures are in close spatial proximity, and lastly (d) *logical*, due to any other mechanism (e.g., policy, legal, or regulatory regimes) that can link logically two or more infrastructures.

Currently, the best way to study the characteristics and operations of multiple interconnected CISs is through accurate modeling and extensive simulation, where interdependencies should be carefully considered [9]. For this reason, several solutions have been proposed in the literature (as discussed extensively in Section 2). However, there is still a need to develop new and more accurate frameworks that will more closely represent component dependencies and handle the

[☆] This work is supported by the European Research Council Advanced Grant FAULT-ADAPTIVE (ERC-AdG-291508).

[*] Corresponding author.
E-mail address: christosp@ucy.ac.cy (C.G. Panayiotou).

http://dx.doi.org/10.1016/j.ress.2017.03.028
Received 5 August 2016; Received in revised form 8 February 2017; Accepted 22 March 2017
Available online 23 March 2017

increasing complexity of larger system (as widely recognized in [3,10]). In accordance, this work introduces a framework based on open hybrid automata for modeling CISs interdependencies by adopting and extending our previous work in [11]. More specifically, the proposed framework uses open hybrid automata for modeling components of different CISs taking into account their dependencies. These models can be composed together to create even larger and more complex models which can then be used in detailed simulations to: i) study the cascading effects of (inter)dependencies, ii) perform vulnerability assessment, and iii) develop maintenance planning strategies. The main advantage of using open hybrid automata is the common modeling framework they provide, where generic CISs components can be modeled. Moreover, open hybrid automata allows the development of models at various levels of abstraction, i.e., very detailed models with many variables or quite simple ones, depending on the modeling objectives and the available data (more details on the modeling abstraction are provided in Section 3).

The important benefit of the proposed approach is that it provides a unified and convenient framework for modeling the various components that make up a critical infrastructure. The modeling framework allows to model both continuous-time and discrete-event dynamics and can easily incorporate decisions made by operators as well as component faults that may occur during operation. The various component models can be reused using composition to build bigger infrastructures and can be connected in various ways to capture different network topologies. Furthermore, the interconnections of the various components seamlessly capture the intradependencies in a critical infrastructure and the interdependencies between infrastructures. The modeling framework can be used to build large scale infrastructures through the model composition and reuse them to build larger models. At the same time, it is flexible to use more or less detailed models depending on the required level of abstraction such that it becomes more scalable during run time.

The rest of the paper is organized as follows. Section 2 summarizes the existing interdependency modeling approaches. Section 3 provides a short introduction to hybrid systems and explains why they are suitable for modeling CISs interdependencies. Section 4 describes the open hybrid automata framework for modeling components of different CISs, and also their composition with respect to their dependencies. Section 5 derives six open hybrid automata component models of three interdependent CISs (i.e., power, telecom and water), and then links them together to create an overall composition model. Section 6 identifies and analyzes the problems that can be tackled by the proposed framework, and Section 7 provides detailed analysis on the insights gained by applying the proposed framework in a simulation setup. Finally, Section 9 provides concluding remarks and future directions for research.

2. Related interdependency modeling approaches

Interdependency modeling is an emerging research field, that includes several innovative modeling approaches. Existing models are summarized and compared in several review works [12,3,13,10,14], making it quite easy to study the state-of-the-art. Among the most popular ones are the input-output methods, agent-based modeling and network based approaches.

Input-output methods are based on the economic equilibrium theory of W. Leontief, and they can estimate at a holistic level the inoperability (i.e., the percentage of malfunction) of infrastructures using the dependency coefficients (also known as Leontief coefficients). However, these coefficients are difficult to calculate correctly, thus they are generally high level approximations following the assumption that interdependencies are related to high economic interaction [15,16].

Agent-based modeling (ABM) approaches take advantage of the fact that CISs can be characterized as complex adaptive systems (CAS) (i.e., complex collection of interacting components that can be altered from learning processes). ABM uses a bottom-up design strategy, and the different CISs components are represented as autonomous agents with attributes, behaviors, and decision-making rules, while interdependencies usually emerge from the agent interactions [17,18].

Network based approaches generally assume that each CIS consists of a set of components (usually represented as nodes) forming a network, and any existing dependencies are represented as relationships between nodes belonging to different networks [19]. Using network-based models for interdependent CISs, it becomes quite easy to perform topological analysis (i.e., describe qualitative connectedness for a set of components). For functional analysis however, network-based models are quite poor. Usually they assume simplifying hypotheses, with functional models able to capture only the basic features of the networks and not the complex effects related to the exact technological implementations [4,20].

There are also several other approaches for modeling CISs interdependencies. For example, there are methodologies based on petri nets, stochastic activity networks and bayesian networks [21–23]. The System Dynamic (SD) approach was also used for interdependency modeling to determine the best allocation strategies from the available infrastructure services when CISs suffer disruptions [24]. Multi-layer modeling approach was also proposed, where infrastructures are seen at different layers (i.e., holistic, service, reductionistic, etc.), with interactions and functional relationships between components and infrastructures modeled at different levels of granularity [25]. Federated simulations using the High Level Architecture (HLA) standard were also used in interdependency modeling studies, with HLA generally acting as communication middleware between different infrastructure simulators, allowing the capture of interdependencies within a “system-of-systems” approach [26,27]. Lastly, empirical approaches have been used to analyze CISs interdependencies according to historical accident or disaster data, and expert experience [8].

So far, the approaches that have appeared in the literature may serve different purposes and have their strengths and weaknesses, but no single approach has become the state-of-the-art of the field. Furthermore, the difficulty in accessing data due to confidentiality and privacy issues, coupled with the fact that CISs are becoming increasingly larger and more complex, makes the validation of interdependencies quite challenging [3,10]. Thus, there is a need to further develop existing interdependency modeling approaches or to propose new ones that are both efficient and effective.

When dealing with critical infrastructures, scalability is an issue faced by all methods that can be used to model interdependencies simply due to the large scale involved in such infrastructures. The proposed approach is both modular and scalable in the sense that it allows for the flexibility of incorporating both highly accurate and simple models in an all-encompassing framework. Modularity is achieved through composition while scalability exists in two forms; scalability in building a model (topology and functionality) of the critical infrastructure as well as scalability in terms of the computational power required to run the model. In terms of modeling, the methodology allows for the composition of multiple models which can be made into higher level components that can be reused to build bigger models. Thus, one does not need to always start from a single component. For example a power plant is a collection of several generators. Thus, one can build the model of one generator and then reuse and connect several generators together to make the power plant. In terms of the needed computational power to run the model, again the methodology allows one to use the appropriate level of modeling abstraction. For example, a switch can be modeled by a simple 0/1 function or, if one requires to also capture the transient effects when a switch opens and closes, these can also be incorporated in the model at the expense of more computational power. The framework provided by hybrid automata allows accurate investigations to be conducted on component dependencies and system interdependencies for studying cascading effects, for vulnerability assessment and for proper planning, as shown in the sequel.

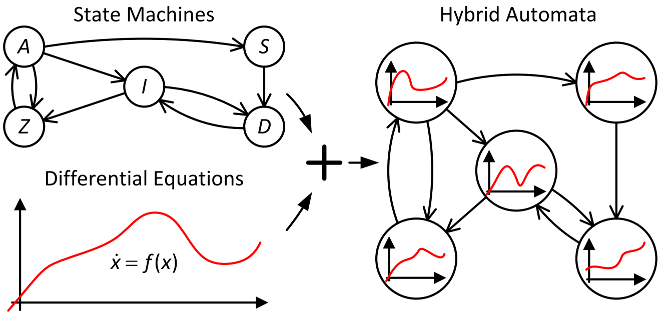

Fig. 1. Hybrid automata integrate state machines and differential equations in a single formalism with uniform mathematical semantics.

3. Hybrid systems

Hybrid systems combine discrete events and continuous time dynamics that can serve as models of large scale systems [28]. The formal models for hybrid systems are called *hybrid automata* and they integrate state machines and differential equations in a single formalism with uniform mathematical semantics (see Fig. 1). There are two types of hybrid automata, *autonomous* and *open*. The difference between the two is that the latter includes inputs and outputs, which makes it possible to model subsystems individually and then compose them together to create more complex and detailed system models.

Over the years hybrid systems have been used to model individual CISs for studying and analyzing their performance. For example, hybrid systems have been used to model and simulate power systems [29,30] to study their behavior under various conditions [31]. A hybrid systems framework was also used to model communication networks to analyze the traffic flow, achieving similar results with sophisticated packet-level simulators (such as ns2) [32]. Furthermore, hybrid systems have been used in transportation systems, modeling and analyzing highway traffic [33]. Finally, hybrid systems were also used to model oil & gas systems. Specifically, in [34] a hybrid system framework was used to model gas transmission networks to investigate their control and management in crisis situations, while in [35] a hybrid system framework was used to model an oil production operation, showing that hybrid systems can provide a more precise description of the process behavior of complex industrial applications.

Nevertheless, apart from modeling individual CISs, hybrid systems can also be used to model multiple interdependent CISs as elaborated in this work. The first step towards that direction is to properly characterize interdependencies since different interpretations exist in the literature, as stressed in [36]. Specifically, in [2] an interdependency is defined as a bidirectional relationship between two CISs through which the state of each CIS influences or is correlated to the state of the other. More generally, two CISs are interdependent when each is somehow dependent on the other. Thus, interdependencies are considered at the level of infrastructures and not between individual components of CISs; as some other interpretations suggest and treat dependencies and interdependencies terms as synonyms (e.g., [8]). In this paper, similarly to [36], bidirectional relationships between infrastructures are modeled as interdependencies, while unidirectional relationships between individual CISs components are modeled as dependencies. This interpretation also correlates with the level of modeling abstraction, which is usually selected with respect to the modeling objectives and the data available.

Modeling abstraction is a relative concept and is considered as the valid simplification of reality and proper reduction of complexity, thus less complexity in a model leads to a higher level of modeling abstraction, and vice-versa [37]. Model complexity can be seen as the "product" of scope and resolution (product is used symbolically), where with scope we refer to how much of the real world is represented (number of components), while with resolution we refer to the number of variables in the model (number of states in each component) and their precision or granularity [38,39]. Thus, in the case of CISs interdependencies, if the selected level of abstraction is high (i.e., each CIS is seen as a single atomic entity with certain resolution) then interdependencies between CISs are considered. However, if the selected level of abstraction is lower (i.e., each CIS is represented with several components and each component has its own resolution), then dependencies between components are considered, with interdependencies formed from dependencies.

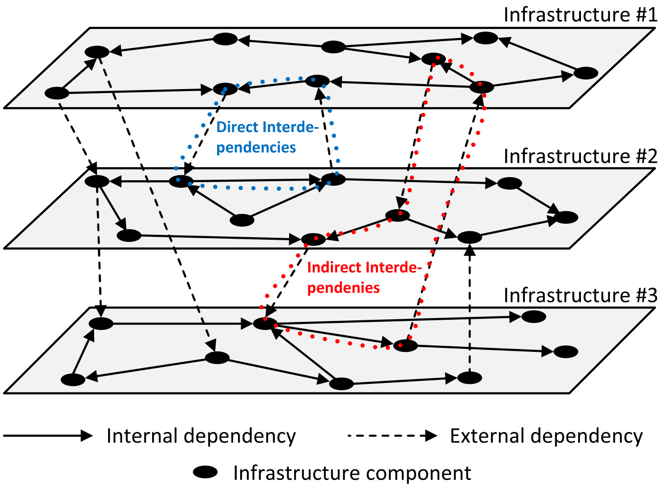

Fig. 2. CISs consist by a large number of components which have internal and external dependencies among them, forming direct and indirect interdependencies between CISs.

In this work each CIS is treated as a collection of components instead of a single atomic entity, with components able to range from parts, units, subsystems, up to systems, depending on the selected modeling abstraction. Each CIS component will be modeled with an open hybrid automaton and any dependencies between components will be represented with the connections between the inputs and outputs of the models. Similarly to [40], two kinds of dependencies are considered, internal and external (as depicted in Fig. 2).

Remark 1. *Internal* dependencies refer to dependencies between components of the same infrastructure (depicted as solid lines in Fig. 2), while *external* dependencies refer to dependencies between components of different infrastructures (depicted as dashed lines in Fig. 2).Because of the number of dependencies between the various components, multiple feedback loops are established among different infrastructures which give rise to interdependencies, either direct or indirect (see Fig. 2).

Remark 2. *Direct* interdependencies denote interdependencies that are created from feedback loops between two CISs due to first order dependencies among their components. *Indirect* interdependencies, on the other hand, denote interdependencies that are created from feedback loops between three or more CISs due to higher order dependencies among their components. Note that indirect inderdependencies are much more difficult to spot and more difficult to analyze than direct ones [36].

By modeling components of different CISs and their dependencies with open hybrid automata, it is possible to represent the various types of interdependencies as well, i.e., physical, cyber, geographic, and logical. Specifically, physical interdependencies can be represented by dependencies between components of different CISs that pass physical commodities (e.g., power, water, oil, etc.). Similarly, cyber interdependencies can be represented by dependencies between components of different CISs that pass information/signals between components (e.g., sensor measurements, control signals, etc.). However, to represent geographic and logical interdependencies we may need extra models other than infrastructure components. For instance, for geographic interdependency, we may need models for earthquake, flood, fire, etc., to drive the different CISs components that are in close spatial proximity to the catastrophic event. Similarly, for logical interdependency, we may need models that represent, for example, human decisions, or policies that may affect components of different CISs. In this work we focus only on modeling CISs components and mostly the first two types of interdependencies, i.e., physical and cyber.

4. Modeling framework

We consider the components of interdependent infrastructures as hybrid systems, and we model each component with an open hybrid automaton. In this paper we adopt the open hybrid automaton framework from [41,42] which is the following collection:

$$\mathcal{H} = (\boldsymbol{Q}, \boldsymbol{X}, \boldsymbol{V}, \boldsymbol{Y}, Init, f, h, Inv, E, G, R) \tag{1}$$

where $q \in \boldsymbol{Q}$ denotes the discrete state and $x \in \boldsymbol{X} \subseteq \mathbb{R}^n$ denotes the continuous state. We let $\boldsymbol{V}$ be a finite set of input variables, which we partition into two subsets $\boldsymbol{V} = \boldsymbol{U} \cup \boldsymbol{\Upsilon}$, where $v \in \boldsymbol{U}$ denotes an internal dependency input and $\hat{v} \in \boldsymbol{\Upsilon}$ denotes an external dependency input. Finally, $y \in \boldsymbol{Y}$ denotes the continuous time output of $\mathcal{H}$.

We refer to the pair $(q, x) \in \boldsymbol{Q} \times \boldsymbol{X}$ as the (hybrid) state of $\mathcal{H}$ with its evolution to be determined by means of (i) initial set of states $Init \subseteq \boldsymbol{Q} \times \boldsymbol{X}$, (ii) a vector field f: $\boldsymbol{Q} \times \boldsymbol{X} \times \boldsymbol{V} \to \mathbb{R}^n$ that describes the continuous state (time driven) dynamics, (iii) a function h: $\boldsymbol{Q} \times \boldsymbol{X} \times \boldsymbol{V} \to \boldsymbol{Y}$ that describes the output $y \in Y$, (iv) an invariant (or domain) set Inv: $\boldsymbol{Q} \to 2^{\boldsymbol{X} \times \boldsymbol{V}}$ that defines the combinations of states and inputs for which continuous evolution is allowed, (v) a collection of edges $E \subseteq \boldsymbol{Q} \times \boldsymbol{Q}$ that represent possible transitions between discrete states, (vi) a guard condition G: $E \to 2^{\boldsymbol{X} \times \boldsymbol{V}}$ at each edge that once true triggers a discrete transition, and (vii) a reset relation R: $E \times \boldsymbol{X} \times \boldsymbol{V} \to 2^{\boldsymbol{X}}$ at each edge that resets the value of $x \in \boldsymbol{X}$ before each discrete transition.

It is often convenient to visualize open hybrid automata using directed graphs, as shown in Fig. 3. Specifically, with each vertex we represent a discrete state $q \in \boldsymbol{Q}$, and inside the vertex we include a vector field $\dot{x} = f(q, x, v)$, an output function $y = h(q, x, v)$, and an invariant set $Inv(q)$. The edges of the directed graph represent discrete transitions. An edge $(q, q') \in E$ starts at $q \in \boldsymbol{Q}$ and ends at $q' \in \boldsymbol{Q}$. For each edge $(q, q') \in E$, we note at the beginning the guard condition $G(q, q')$ that once true triggers a discrete transition, and at the end the reset relation $R(q, q', x, v)$ only if the continuous state resets to a new value. The set of initial states $Init$ is visualized with an arrow that points at the initial discrete state q_0 and also specifies the initial value of the continuous state e.g., $x = x_0$. Finally, the input variables $\boldsymbol{V}$ are added on the left, while the output variables $\boldsymbol{Y}$ on the right.

A sample path or execution of an open hybrid automaton consists of a sequence of intervals of continuous evolution followed by a discrete transition. The execution starts from some initial state $(q_0, x_0) \in Init$. The model remains at a discrete state q as long as the continuous state $x \in \boldsymbol{X}$ and/or the input $v \in \boldsymbol{V}$ does not leave the set $Inv(q)$. At the same time, the output $y \in \boldsymbol{Y}$ is determined by $h(q, x, v)$. If x and/or v reach the guard condition $G(q, q')$, then a discrete transition to $q' \in \boldsymbol{Q}$ takes place instantaneously, with the value of continuous state x determined by the reset relation $R(q, q', x, v)$. The continuous evolution then resumes to the new discrete state q'. The aforementioned procedure repeats accordingly. More formally, the execution of an open hybrid automaton is defined as a collection $\chi = (\tau, q, x, v, y)$, consisting of the *hybrid time set* $\tau \in \mathcal{T}$ and q: $\tau \to \boldsymbol{Q}$, x: $\tau \to \boldsymbol{X}$, v: $\tau \to \boldsymbol{V}$, and y: $\tau \to \boldsymbol{Y}$ satisfying the initial conditions, continuous and discrete evolution, and also output evaluation. More extensive definitions and the theory for open hybrid automata can be found in [28,42,43].

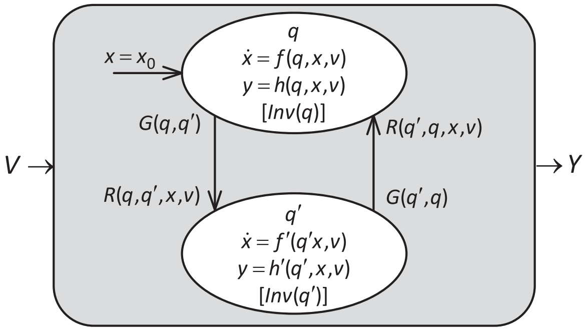

Fig. 3. Graphical representation of open hybrid automaton.

Because we represent each CIS component with an open hybrid automaton, it is necessary to be able to connect all these models together properly and create a larger model that we can use to simulate different scenarios. This can be achieved with the composition operation of open hybrid automata [42–44]. In general, the composition of two open hybrid automata models consists of, roughly, the connection between some of the inputs/outputs (I/O) of one hybrid automaton with some O/I of another, while the remaining I/O become the I/O of the newly composed hybrid automaton [41].

In our case, where we compose models of components of different interdependent CISs, the connections between inputs and outputs represent internal and external dependencies, which can become considerably large very quickly. To address this, specific notation is added to input and output variables that makes the composition clearer and more meaningful. Recall that for a CIS component model, we denote with $v \in \boldsymbol{U}$ an internal dependency input and with $\hat{v} \in \boldsymbol{\Upsilon}$ an external dependency input, while with $y \in \boldsymbol{Y}$ we denote an output for the model. In all these variables we add superscripts that indicate a particular infrastructure and subscripts that indicate the specific component of a particular infrastructure. For example, $\hat{v}^s_{n,m}$ denotes an external dependency input m of component n at infrastructure system s. Fig. 4 provides a visual example of the composition of four components, two for infrastructure "a" and two for infrastructure "b", using the aforementioned notation for the input and output variables.

As elaborated above, using the open hybrid automata framework a plethora of CISs component models can be developed at the necessary level of abstraction. Subsequently, exploiting the open hybrid automata composition properties, the component models can be connected together based on the dependencies among them to reflect a variety of dynamic systems. Furthermore, the composition model is modular and scalable, since component models can be added/upgraded or replicated as necessary based on the modeling objectives.

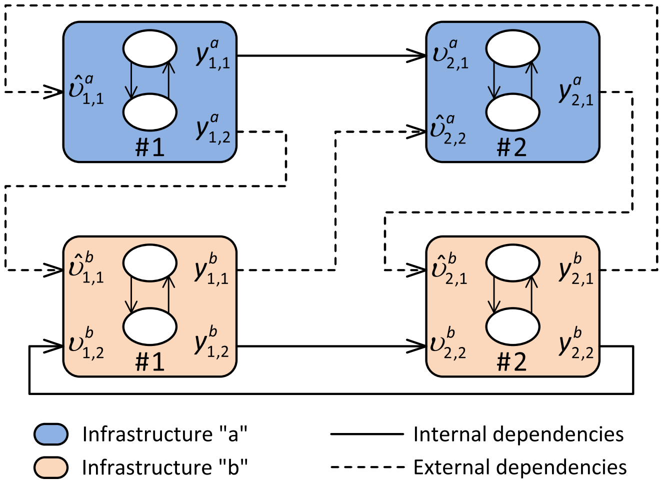

Fig. 4. Composition example of open hybrid automaton models of CISs components with the notation for input and output variables.

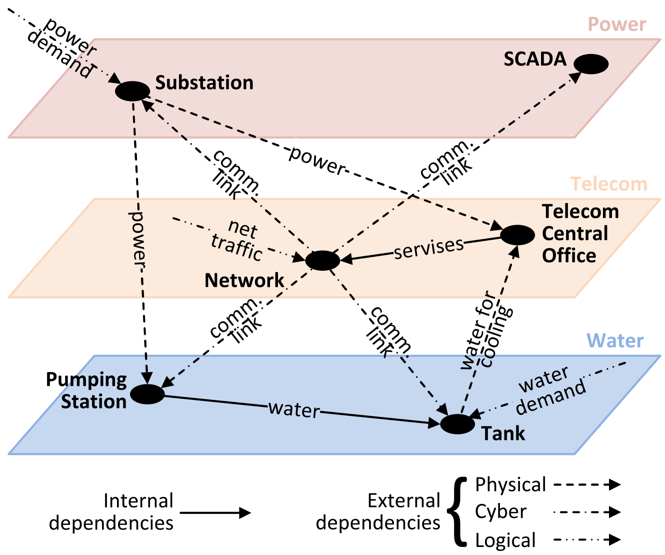

Fig. 5. Overview of CISs components and their dependencies.

5. CISs component models

In the sequel, the open hybrid automaton framework is used to model components that can be found in a small city and have dependencies between them, from three different CISs, namely power, telecom, and water. Specifically, we model the following six components: for the power system (i) a distribution substation and (ii) a SCADA system that monitors the substation; for the telecom system (iii) a telecom central office and (iv) a telecommunication network; and for the water system (v) a water tank and (vi) a pumping station.

As depicted in Fig. 5, there are several internal and external dependencies between the various components. The identified internal dependencies are (i) the dependency of the tank on the pumping station for water supply and (ii) the dependency of the network on the telecom central office for switching and routing. Three types of external dependencies have also been identified: physical, cyber, and logical. Physical dependencies include (i) the dependency of the telecom central office and the pumping station on the substation for power supply, and (ii) the dependency of the telecom central office on the tank for water supply. Cyber dependencies exist for all network communications links, i.e., the links to the substation, the SCADA, the pumping station, and the tank. Lastly, there are logical dependencies, such as the power demand for the substation, the water demand for the tank, and the data traffic for the communication network. These logical dependencies are directly related to human behavior (e.g., consumer demand on electricity, water, and communication).

The goal is to demonstrate the application of the proposed framework rather than create a low abstraction (i.e., high complexity) models for each CIS component. Thus, we designed models for only six components at a reasonable level of abstraction to keep the composition model simple and understandable but at the same time complex enough to serve its purpose. To aid with readability of the models, the input and output variables outside each model, that use the generic notation described in Section 4 (and illustrated in Fig. 4), are mapped with variables inside the model that are more related to the operation of each CIS component.

In the following subsections the open hybrid automata models of the components that make up the power infrastructure are explained in detail. Furthermore, a brief description is provided for the models of the components of the water and telecommunications infrastructures. The full description of the components of these infrastructures can be found in A and B respectively. Also, Table 1 summarizes all the model variables for quick reference, while Section 5.4 derives the composition model.

Table 1
Summary of the models' variables.

Power Substation			
s_{CB}	SCADA signaling	x_{ti}	time inverse relay
p_d	power demand	i	line current
$l_{OP.}$	local operator input	I_{lim}	substation limit current
p_s	power supply	V_{nom}	substation nominal voltage
p_m	sensor power measurement	k	delay parameter
		ε	measurement noise
Power SCADA			
p_m	sensor power measurement	x_t	timer
s_{OP}	operator input	T_d	max transmission delay
s_{CB}	substation control signal		
Water Tank			
w_d	output water demand rate	x_v	tank volume state
w_s	water supply input rate	V_{max}	maximum volume
v_{tank}	tank volume measurement	V_0	initial volume
w_{out}	output water supply rate		
Water Pumping Station			
v_{tank}	tank volume measurement	P_0	station power demand
p_s	power supply to station	T_{off}	station resting period
ϕ_P	technical fault	T_{on}	station working period
w_s	output water supply rate	V_{max}	tank maximum volume
p_d	power demand of station	V_{th}	volume threshold
x_t	timer for station operation	W_{avg}	avg. water supply rate
Telecom Central Office			
p_s	power supply to office	x_{ups}	timer for UPS operation
w_s	water supply for cooling	x_{wtr}	timer for cooling system
ϕ_T	technical fault	T_{ups}	UPS availability period
z_{tco}	office operation status	T_{wtr}	cooling system availability period after water shortage
p_d	power demand of office		
w_d	water demand of office	P_0	office power demand
		W_0	office water demand
Telecommunications Network			
s_1	SCADA to subst. ctrl. signal	q	aggregated queue size
s_2	tank to station volume signal	r	transmission rate
s_3	subst. to SCADA power measurement signal	B	network bandwidth
		B_ϕ	reduced network bandwidth
p	num. of input packets	a	association parameter
ϕ_l	number of failed links	T_p	propagation delay
z_{tco}	office operation status	P_k	packets num. per signal k

5.1. Power infrastructure components

5.1.1. Power distribution substation

The main task of a distribution substation is to supply power to the city, including the telecom central office, and the water pumping station. The substation is monitored remotely by the SCADA, and is equipped with overcurrent protection based on inverse time relay, similar to the setup in [30].

We model the substation with the open hybrid automaton as shown in Fig. 6. The model has two discrete states, (i) "Power Supply" for when the substation operates normally and, (ii) "CB Open" for when the circuit breaker (CB) opens and cuts the power supply. While the model is in the "Power Supply" state the two outputs, p_s (that denotes the power supply) and p_m (that denotes the power sensor measurement signal), become p_s=p_d and $p_m = p_d + \varepsilon$, where p_d is the input that denotes the power demand and ε the white noise in the measurements. The continuous state of the model x_{ti} represents the time inverse relay for detecting overcurrents which is calculated by $\dot{x}_{ti} = i - I_{lim}$, where I_{lim} is the substation's limit current and i is the line current that is given by $i = p_d/V_{nom}$, with V_{nom} denoting the substation's nominal voltage. The value of x_{ti} changes with respect to the line current i. Initially, $x_{ti} = -k < 0$, where k is the delay parameter and the lower saturation limit for x_{ti}. If i is larger than I_{lim}, then x_{ti} will increase until $x_{ti} \geq 0$ when overcurrent will occur and the guard will trigger a

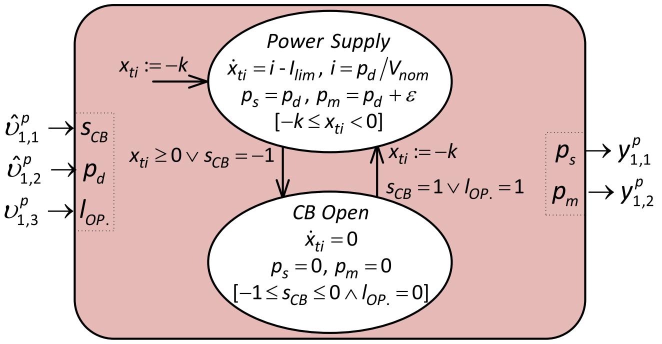

Fig. 6. Open hybrid automaton for the power distribution substation.

discrete transition to the "CB Open" state. The same transition is also triggered if the input s_{CB}, which represents the SCADA signaling, becomes $s_{CB} = -1$, denoting the remote opening of the circuit breaker. A discrete transition from the "CB Open" back to the "Power Supply" state will be triggered if the input s_{CB} becomes $s_{CB} = 1$, which represents the SCADA signaling for the remote closing of the circuit breaker. In case that remote closing is not possible, due to communication issues, a local operator can close the circuit breaker, which is denoted with the input $l_{OP.} = 1$.

5.1.2. Power SCADA

The power SCADA remotely monitors and controls the power substation by receiving sensor measurements and sending control signals through the communication network. We model the SCADA with the open hybrid automaton as shown in Fig. 7. The discrete states of the model are: (i) "CB Close" and (ii) "CB Open" both representing the status of the substation's circuit breaker, (iii) "Send CMD" when the SCADA sends commands to the substation, and (iv) "Conn. Down" when the network is down. The transitions between the discrete states are triggered by the guards that use: the input p_m that denotes the sensor power measurements, the input s_{OP} that denotes the signal from the operator, and the continuous state x_t that acts as a timer for the transmission delay. Specifically, some discrete transitions are triggered only by the p_m input value. For example, if p_m=0 there will be a transition to the "CB Open" state, if $p_m > 0$ there will be a transition to the "CB Close" state, and if $p_m = NaN$ (i.e., Not a Number) there will be a transition to the "Conn. Down" state. Other discrete transitions are triggered by the s_{OP} input values. For example, there will be a transition to state "Send CMD" from states "CB Close" and "CB Open" if $s_{OP} = -1$ and $s_{OP} = 1$, respectively. For both these transitions, the continuous state x_t will reset to x_t=T_d, where T_d denotes the maximum transmission delay, and once in the "Send CMD" state x_t will be used as a timer i.e., $\dot{x}_t = -1$. Also, while in this state the output s_{CB}, which denotes the signal that is sent to the substation, becomes $s_{CB} = s_{OP}$, i.e., $s_{CB} = -1$ for opening the circuit breaker and $s_{CB} = 1$ for closing the circuit breaker. Finally, once x_t=0, and based on the value of p_m (i.e., $p_m > 0$ or p_m=0), a transition will be triggered to either the "CB Close" or "CB Open" states.

5.2. Water infrastructure components

The water infrastructure consists of a tank and a pumping station. The tank supplies the city with water according to the demand and receives water from the pumping station in a controlled way to avoid drainage or overflow. The level of the water in the tank is controlled by the pumping station which receives measurements for the tank's water level and compares it to an appropriate threshold. Furthermore, the model includes pump constraints such as maximum pumping, maximum operating period and minimum resting period which affect the water supply of the tank. The models also capture communication network problems, power cuts, and technical faults, which affect the proper operation of the pumping station. Detailed descriptions for both water infrastructure components can be found in Appendix A.

5.3. Telecommunications infrastructure components

The telecommunications infrastructure consists of a telecom central office and a telecommunications network. The telecom central office houses all the critical network equipment, such as switches and servers, as well as support systems, such as cooling systems and uninterruptible power supply (UPS) to ensure that the office equipment continues to function even during power cuts. The telecom central office controls the telecommunications network which consists of nodes connected with links characterized by finite bandwidth. The network is continuously loaded with packets that are generated at the nodes and sent through the links. The packet transmission rate is determined by the network traffic (number of packets in the network) and the queue size at the links that temporarily hold packets before transmission. The network can suffer from link failures, that reduce the bandwidth and increase latency. Moreover, the network can go down if there is a failure at the telecom central office, since necessary services, such as switching and routing, are no longer available. Detailed description for both telecommunication infrastructure components can be found in Appendix B.

5.4. Composition

The six open hybrid automata models are composed together as shown in Fig. 8, creating a larger model that includes the various dependencies between the components. As elaborated above, there are a few internal dependencies and a number of external dependencies of various types (physical, cyber and logical), that are depicted in detail in Fig. 8.

In the composition all models run in parallel, and, based on the dependencies between them, the output of one model becomes an input to the other. In this way, various feedback loops are developed between

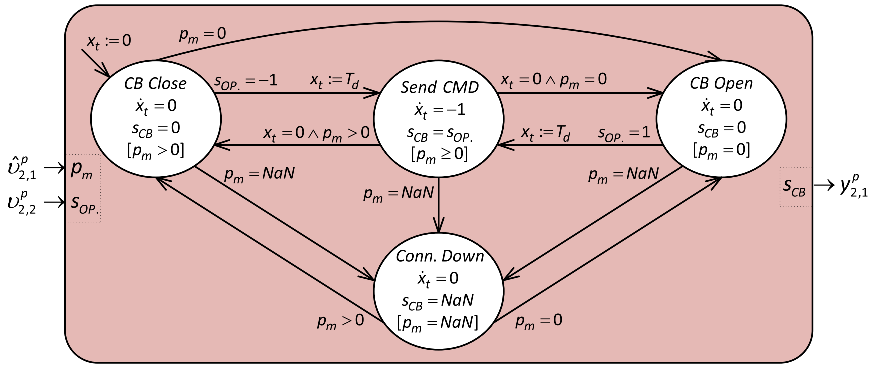

Fig. 7. Open hybrid automaton for the power SCADA.

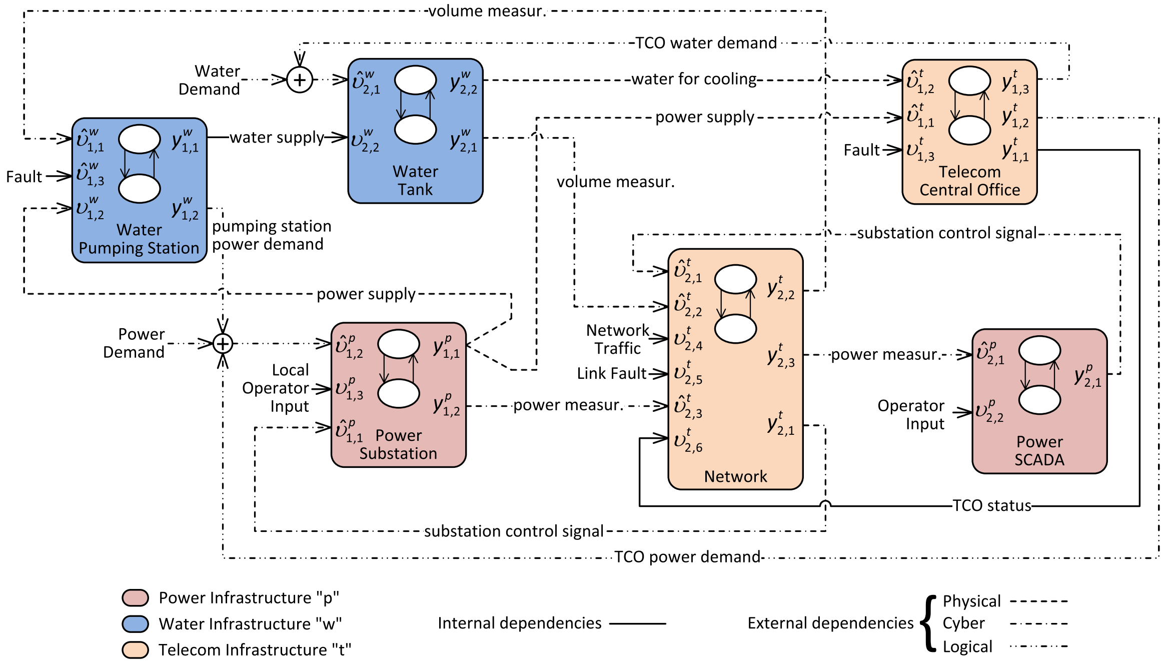
volume measur.
TCO water demand
Water Demand
water for cooling
water supply
power supply
Fault
$\hat{\upsilon}^w_{1,1}$ $\upsilon^w_{1,3}$ $\upsilon^w_{1,2}$ $y^w_{1,1}$ $y^w_{1,2}$
Water Pumping Station
$\hat{\upsilon}^w_{2,1}$ $\upsilon^w_{2,2}$ $y^w_{2,2}$ $y^w_{2,1}$
Water Tank
$\hat{\upsilon}^t_{1,2}$ $\hat{\upsilon}^t_{1,1}$ $\upsilon^t_{1,3}$ $y^t_{1,3}$ $y^t_{1,2}$ $y^t_{1,1}$
Telecom Central Office
volume measur.
pumping station power demand
substation control signal
power supply
Power Demand
Local Operator Input
$\hat{\upsilon}^p_{1,2}$ $\upsilon^p_{1,3}$ $\hat{\upsilon}^p_{1,1}$ $y^p_{1,1}$ $y^p_{1,2}$
Power Substation
Network Traffic
Link Fault
$\hat{\upsilon}^t_{2,1}$ $\hat{\upsilon}^t_{2,2}$ $\upsilon^t_{2,4}$ $\upsilon^t_{2,5}$ $\hat{\upsilon}^t_{2,3}$ $\upsilon^t_{2,6}$ $y^t_{2,2}$ $y^t_{2,3}$ $y^t_{2,1}$
Network
power measur.
power measur.
Operator Input
$\hat{\upsilon}^p_{2,1}$ $\upsilon^p_{2,2}$ $y^p_{2,1}$
Power SCADA
TCO status
substation control signal
TCO power demand
Power Infrastructure "p"
Water Infrastructure "w"
Telecom Infrastructure "t"
Internal dependencies
External dependencies
Physical
Cyber
Logical

Fig. 8. Model composition of CIS components.

the models, which represent interdependencies. For instance, there are connections between the inputs and the outputs of the power substation model and the power SCADA model that pass through the network model, since they use it to transmit power measurements and control signals, respectively. These dependencies create feedback loops between these models, which subsequently form cyber interdependencies. Thus, in case the network fails, it will affect the other two models as well.

Lastly, note that the model composition derived in this work is quite scalable. Components can be added/upgraded or replicated as needed depending on the modeling objectives and the data available. Of course, more component models means a more complex model composition and a more accurate model that contains more dependencies, but also implies more computational power.

6. Interdependency analysis

Having a composition model as the one in Fig. 8, it becomes possible to run simulations with various objectives. Firstly, simulations can be used to provide information about the consequences that a loss of components in one infrastructure can have to other infrastructures. In more detail, it is possible to run scenarios where components in one infrastructure fail, and then observe whether this failure can affect components in other infrastructures, due to cascading and higher order effects because of dependencies and interdependencies. As an example, Fig. 9 shows the possible cascading effects that can occur after a fault at the telecom central office.

Secondly, we can use the composition model for vulnerability analysis, where different scenarios can be executed to investigate vulnerabilities on specific infrastructure components. For example, we can run several simulations with specific components intentionally failing at different times and for different durations in order to observe how components in other infrastructures are affected as a result.

Finally, the composition model can be used for planning, i.e., determine when it is best to plan maintenance, repair, or upgrade of specific components. Doing so, allows one to plan the extend of downtime that the system can tolerate, and thus the goal is to determine when it is the best period that these down-times should occur, so that there is a minimum disturbance to components of other infrastructures.

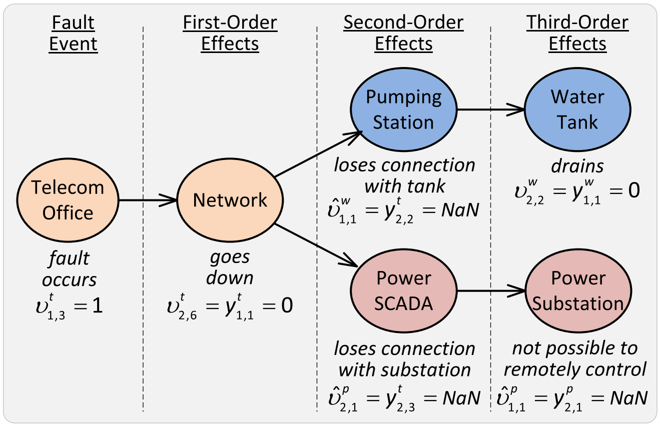
Fault Event
First-Order Effects
Second-Order Effects
Third-Order Effects
Telecom Office
fault occurs
$\upsilon^t_{1,3}=1$
Network
goes down
$\upsilon^t_{2,6}=y^t_{1,1}=0$
Pumping Station
loses connection with tank
$\hat{\upsilon}^w_{1,1}=y^t_{2,2}=NaN$
Power SCADA
loses connection with substation
$\hat{\upsilon}^p_{2,1}=y^t_{2,3}=NaN$
Water Tank
drains
$\upsilon^w_{2,2}=y^w_{1,1}=0$
Power Substation
not possible to remotely control
$\hat{\upsilon}^p_{1,1}=y^p_{2,1}=NaN$

Fig. 9. Example of n-order cascading effects due to dependencies, after a fault in the telecom central office.

7. Simulation results

We use the composition model in Fig. 8 to illustrate the capabilities of the proposed hybrid systems approach with respect to the different objectives discussed in Section 6. In the numerical analysis that follows, we have used with the composition model the parameter values shown in Table 2 and the input signals for power and water demand shown in Fig. 10. All models have been implemented in Matlab/Simulink software [45] and simulated using a machine with the following specifications: Intel Core i5-2400 3.1 GHz; 4 GB RAM; Windows 7; Matlab R2014a.

We first simulate the composition model to obtain information for possible consequences that a loss of a component in one infrastructure can have on other CISs components, due to cascading and higher order effects because of (inter)dependencies. Figs. 11 and 12 present the results for a 24 h scenario. Specifically, Fig. 11 shows a timeline with the induced events/faults (at the top of the figure), and their subsequent consequences (at the bottom of the figure). Fig. 12 shows plots for each component's performance during the scenario, while the numbers depicted at the two figures associate the events in the timeline with each plot.

Table 2
Model parameter values used in the simulations.

Power Substation	
power demand: current limit:	p_d (see Fig. 10(a)) I_{lim} = 1 pu (per unit)
nominal voltage: delay parameter:	V_{nom} = 1 pu k=2
measurement noise:	$\varepsilon = unif(-0.05, 0.05)$
Power SCADA	
transmission delay:	T_d = 60 s
Water Tank water demand:	 w_d (see Fig. 10(b))
initial volume:	V_0 = 150 m^3
maximum volume:	V_{max} = 1000 m^3
Water Pumping Station	
volume threshold:	V_{th} = 100 m^3
resting period:	T_{off} = 1 h
working period:	T_{on} = 2.5 h
average supply rate:	W_{avg} = 0.15 m^3/s
tank max volume:	V_{max} = 995 m^3
power demand:	P_0 = 0.01 pu
Telecom Central Office	
UPS operat. time:	T_{ups} = 1 h
cooling availability:	T_{wtr} = 2 h
power demand:	P_0 = 0.01 pu
water demand:	W_0 = 0.005 m^3/s
Telecommunications Network	
bandwidth:	$B = 1250 packets/s$
propagation delay: association factor:	T_p = 0.04 s a=50
input packets:	$p = poisson(\lambda = 35)$
signal packets:	$P_{k=1} = unif(5, 50)$ $P_{k=2} = unif(10, 20)$ $P_{k=3} = unif(2, 60)$

From the results we can see that the model can provide information for possible cascading and higher order effects because of (inter) dependencies. For instance, the induced fault (#2) at the telecom central office between 05:15-06:04, immediately forces the communication network to go down (#2 a), which then causes the SCADA to lose communication with the substation (#2 b). The pumping station also loses connection with the water tank, which keeps supplying the tank with water, resulting eventually to an overflow at 06:00 (#2 c). Another example is the remote opening of the substation circuit breaker at 21:26 (#5). This immediately causes the telecom office to make use of the UPS (#5 a), and the pumping station to go out of service (#5 b). Because the circuit breaker remains open for an extended period, the telecom office UPS system depletes after one hour (#5 c), and the telecom office stops operating. This causes the network to go down (#5 d), and subsequently the power SCADA to lose communication with the substation (#5 e). Since the SCADA operator is not able to remotely close the circuit breaker (#5 f) due to the telecommunication failure, and a local operator in the substation is not available in this scenario, the simulation goes into a deadlock; on the one hand, the substation waits for the remote closing command to come from the SCADA through the network, while on the other hand the network waits for the telecom central office to power up from the substation.

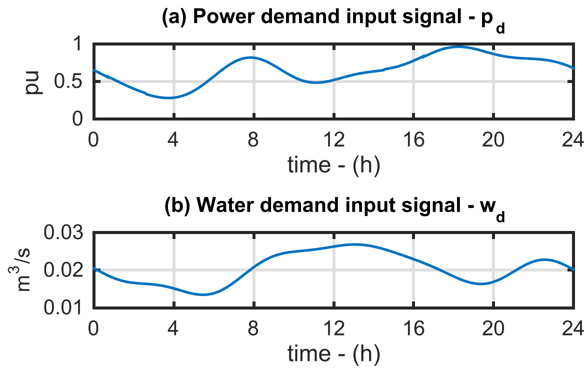

Fig. 10. Profiles of input signals for power demand (p_d) and water demand (w_d) for all simulations.

In addition to investigating cascading events, the composition model can be employed for vulnerability analysis. This is demonstrated hereafter, by investigating how the duration of a fault at the pumping station, and a fault at the telecom central office, may affect other CISs components. To accomplish this, two different simulation scenarios are considered, one where a fault at the pumping station is induced (i.e., $\phi_p = 1$ in Fig. 17) and one where a fault at the telecom office is induced (i.e., $\phi_T = 1$ in Fig. 18). The duration of the induced faults follows a normal distribution, with a mean value 1–10 h at 1 h increments (i.e., a total 10 increments considered), and with variance of 15 min. The fault's starting times are determined randomly at the beginning of the simulation uniformly between 0–14 h, so that the largest fault is able to unfold in the 24 h scenario, while common random numbers are used between the 10 different durations. For each fault, a total of 500 Monte Carlo simulations were executed, with the objective to calculate the mean duration of every other component transitioning to some undesired discrete states. Figs. 13 and 14 show the results for the pumping station fault and for the telecom central office fault, respectively.

With the current setup, from Fig. 13 we can see that if the duration of the pumping station fault is up to 3 h it affects only the tank, and if is up to 4 h it affects the telecom central office, the network, and the SCADA, with only exception the substation that remains unaffected. As the duration of the pumping station fault increases (>4 h), the duration that all aforementioned affected components remain to some undesired discrete state increases as well. Similarly, from Fig. 14 we can see that a fault at the telecom central office immediately starts to affect the network, the pumping station and the SCADA, and if the duration of the telecom office fault is larger than 2 h the tank is affected too, with only exception the substation that remains again unaffected. Comparing the results from the pumping station fault in Fig. 13 and the telecom office fault in Fig. 14 for the current simulation setup (i.e., same parameter values and input signals) we can say that a fault at the telecom office is more severe than a fault at the pumping station, since the mean duration that other components are affected is larger.

Finally, the composition model can effortlessly be used for planning purposes. For instance, it can be used to determine when is the best time with the current setup to carry out 2–5 h maintenance at the power substation so that the disruption to all other components is

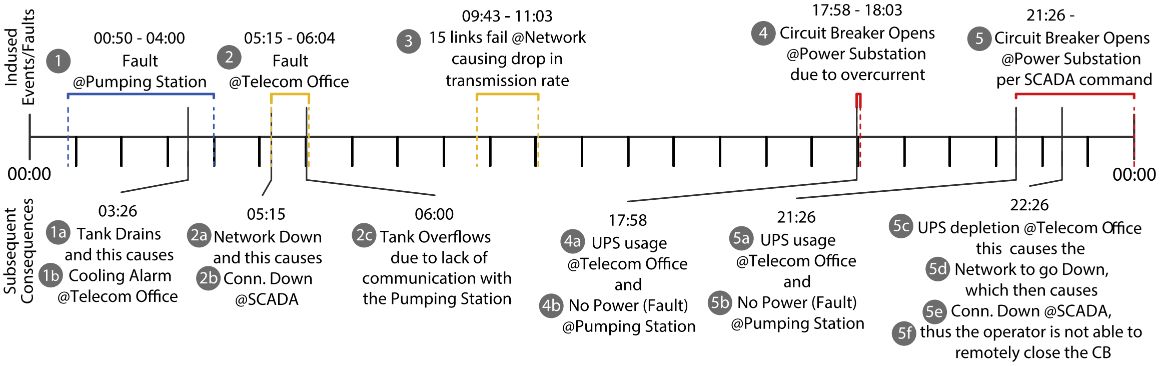
Indused Events/Faults
1 00:50 - 04:00 Fault @Pumping Station
2 05:15 - 06:04 Fault @Telecom Office
3 09:43 - 11:03 15 links fail @Network causing drop in transmission rate
4 17:58 - 18:03 Circuit Breaker Opens @Power Substation due to overcurrent
5 21:26 - Circuit Breaker Opens @Power Substation per SCADA command
00:00
00:00
Subsequent Consequences
03:26 1a Tank Drains and this causes 1b Cooling Alarm @Telecom Office
05:15 2a Network Down and this causes 2b Conn. Down @SCADA
06:00 2c Tank Overflows due to lack of communication with the Pumping Station
17:58 4a UPS usage @Telecom Office and 4b No Power (Fault) @Pumping Station
21:26 5a UPS usage @Telecom Office and 5b No Power (Fault) @Pumping Station
22:26 5c UPS depletion @Telecom Office this causes the 5d Network to go Down, which then causes 5e Conn. Down @SCADA, 5f thus the operator is not able to remotely close the CB

Fig. 11. Simulation timeline with induced events/faults at the top and their subsequent consequences at the bottom.

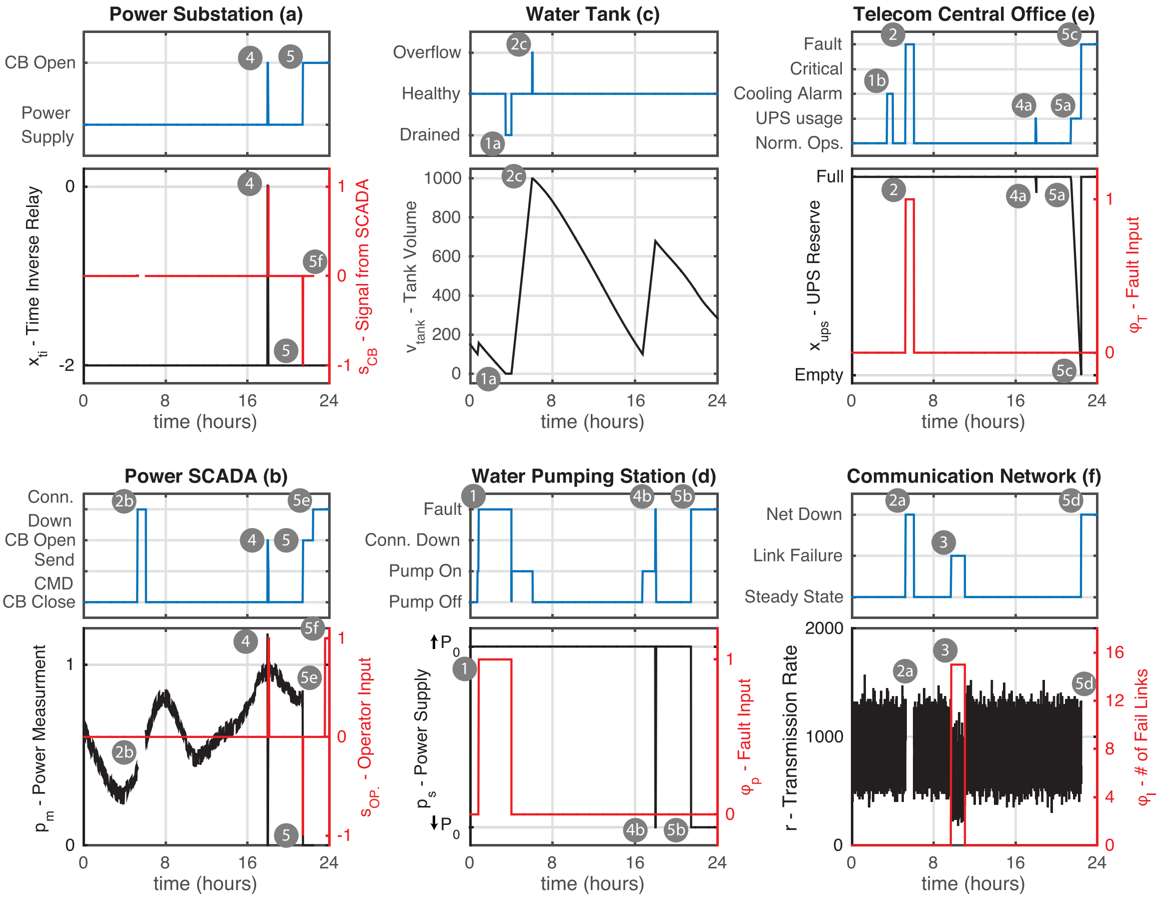

Fig. 12. Plots showing the simulation of fault events for each component associated with the timeline in Fig. 11. For each component we show a plot with discrete state over time and a plot with what is best for each scenario, such as input/output variables and/or continuous state over time.

minimal. To demonstrate this application, we use the composition model to run simulations with the substation circuit breaker opening ($s_{OP.} = -1$) and closing ($s_{OP.} = 1$) for varying lengths of time between 0:00 h and 18:00 h with 30 min increments for a period of 2–5 h with 15 min increments, representing this way all possible substation maintenance strategies. The collected simulation data is used to calculate the total duration that other components are affected (i.e., the total time that they transition to an undesired state) due to the substation maintenance. Fig. 15 shows a plot with the results, where we observe that the substation maintenance should be avoided between 00: 00 – 01: 00 and 11: 30 – 15: 00, since during those times all other components are affected due to the maintenance significantly more than during any other times.

8. Discussion

The purpose of the example presented in this paper is just to demonstrate how the proposed modeling framework can be utilized and provide results related to performance metrics of the infrastructures under investigation. The example exhibits some interesting behaviors such as cascading effects but it is also simple enough for the readers to follow. Clearly other modeling frameworks can be used to model the specific example. As mentioned earlier, the advantage of the proposed methodology is that it provides a convenient framework for modeling continuous and discrete event dynamics as well as decisions, or faulty states and allows for modeling different topologies as well as various types of interdependencies.

Compared to the other methodologies we believe that the proposed hybrid systems approach provides a unified and convenient framework

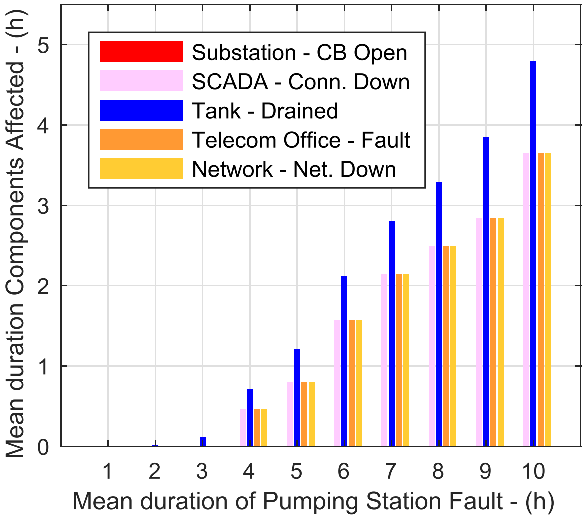

Fig. 13. Plot showing how the duration of a water pumping station fault affects all the other CISs components.

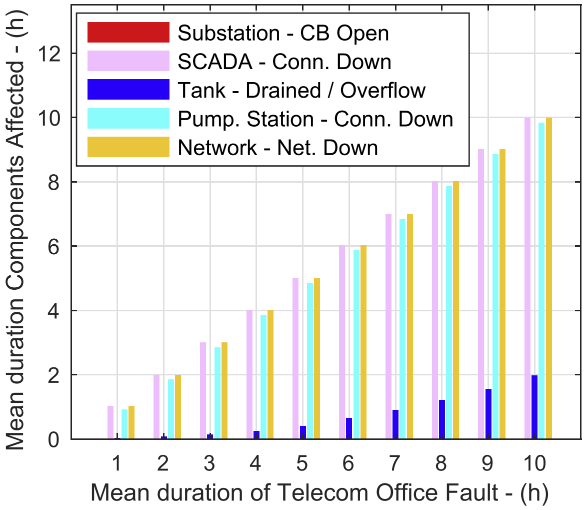

Fig. 14. Plot showing how the duration of a telecom central office fault affects all the other CISs components.

in the form of open hybrid automata for modeling the behavior (dynamic and functional) and topology of any type of interdependent infrastructure system and/or their constituent components at various levels of abstraction. The key advantage of the proposed methodology is its flexibility in the sense that it can use composition to build hierarchical models of components that include both continuous time as well as event driven dynamics. Then, these components can be reused and can be connected to build bigger components and so on. At the same time, each component can be modeled to the appropriate level of abstraction depending on the application needs and the available computing power. The main reason for this is because open hybrid automata combines two popular modeling approaches, namely finite state automata and differential equations, into a single framework and also enables inputs and outputs to the models. Thus, all the discrete states, under operational or faulty conditions, including continuous time dynamics at each discrete state, can be modeled in a convenient framework. Moreover, the inputs and outputs of the models in a composition setup can transfer changes in the behavior of one component to other components, conveniently incorporating the system topology and the interdependencies.

The proposed framework is different from the general network-based approach which is usually used to carry out topological analysis rather than functional, since it mainly represents different CISs components as nodes while links mimic the physical and relational connections among them, with each node or link usually having two discrete states: failed and normal. The network flows method, is another network based approach used to capture interdependencies among CIS, that models some of the CIS functionality by incorporating different types of nodes (i.e., supply nodes, demand nodes or transshipment nodes) and adds to the links capacity limitations to represent the flow of services between infrastructures. With this method, however, the functional behavior that can be represented is quite limited.

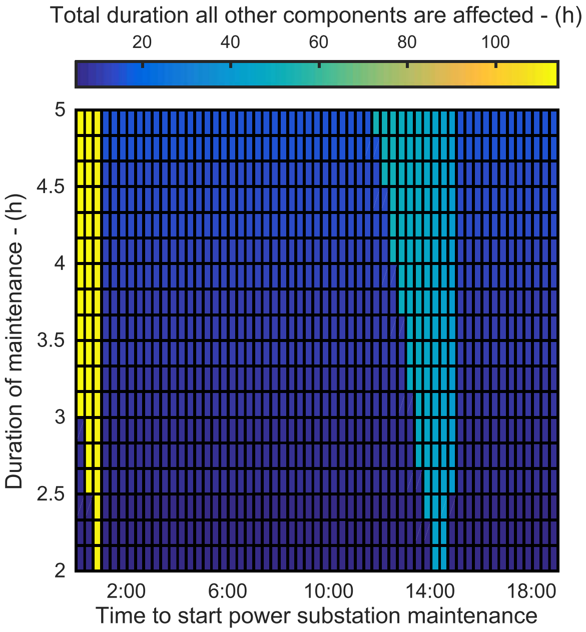

Fig. 15. Plot showing the total duration that CISs components are affected with respect to the start time and the duration of a planned maintenance at the substation.

9. Conclusion and future work

In this work we propose the use of open hybrid automata for developing models of different infrastructure components. These models can be composed together, based on the dependencies that exist between them, and create a larger model that we can use for interdependency analysis, such as investigating cascading effects, performing vulnerability assessment, and planning for maintenance strategies.

Models for six components from three different CISs have been designed and composed together to create a setup that incorporates all the dependencies between the various components. We then use this setup to run simulations that analyze all the aforementioned objectives, with the results able to demonstrate the effectiveness of the proposed modeling framework for interdependency analysis.

In the future we plan to use the proposed approach to develop models with different abstraction levels for various CISs components, having as an end goal the creation of a library where models can be selected and can be easily reused for various studies, incorporating geographical and logical interdependencies as well. We also plan to investigate ways to easily generate scenarios for large composition models, as well as ways to process the data that are produced by these models. Finally, we plan to explore reachability analysis methods for these models, which can help to discover sets of inputs that can send the composition model to undesirable states.

Appendix A. Water infrastructure components

A.1. Water tank

The tank is modeled with the open hybrid automaton shown in Fig. 16. The continuous state x_v denotes the volume of the tank and changes according to $\dot{x}_v = w_s - w_d$, where w_s denotes the input water supply rate from the pumping station, and w_d the output water demand rate. The volume x_v also determines the discrete state of the model. As shown in Fig. 16, the model will remain at the “Healthy” state while $0 < x_v \le V_{max}$, where V_{max} is the tank's maximum volume, and it will transition to either the “Drained” state if $x_v \le 0$, or the “Overflow” state if $x_v > V_{max}$. The model will return back to the “Healthy” state if, while in the “Drained” state, the water supply becomes larger than the demand $w_s > w_d$, or if, while in the “Overflow” state, the water demand becomes larger than the supply $w_d > w_s$. Finally, the two outputs of the model, v_{tank} (that denotes the tank's volume measurement), and w_{out} (that denotes the tank's output water supply rate), take the proper values in each discrete state as shown in Fig. 16.

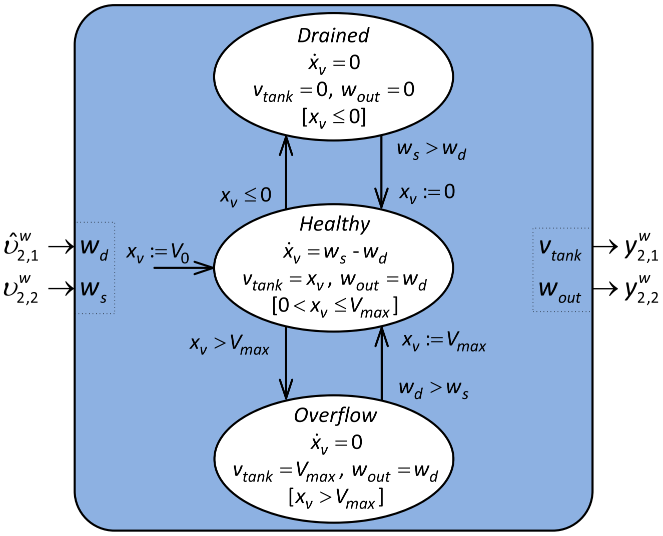

Fig. 16. Open hybrid automaton for the water tank.

A.1.1. Water pumping station

The pumping station is modeled with the open hybrid automaton shown in Fig. 17. The model has four discrete states: (i) “Pump Off” when the station is off, (ii) “Pump On” when the station supplies the tank with water, (iii) “Conn. Down” when the network in unavailable, and (iv) “Fault” when the station is non-operational, due to power cut or technical fault. The discrete transitions between the first two states, “Pump Off” and “Pump On”, are triggered by the input v_{tank}, that denotes the tank's volume measurement, and by the continuous state x_t, that acts as a timer counting the working and the resting period of the pumps. Specifically, the transition from “Pump Off” to “Pump On” is triggered when the water level in the tank goes below the volume threshold V_{th} (i.e., $v_{tank} < V_{th}$) and also the resting period T_{off} for the pumps has expired (i.e., $x_t \le 0$). While the model is in the “Pump On” state, the two outputs of the model, w_s (that denotes the water supply rate), and p_d (that denotes the power demand of the station), become $w_s = W_{avg}$ and $p_d = P_0$, with W_{avg} denoting the average water supply rate and P_0 denoting the necessary power demand of the station while in operation. The transition from the “Pump On” back to the “Pump Off” state, however, is triggered by the guard condition, either when the tank's water volume reaches the maximum volume V_{max} (i.e., $v_{tank} \ge V_{max}$) or if the working period T_{on} of the pumps has been reached (i.e., $x_t \le 0$). Other discrete transitions to the model include the transition to the “Conn. Down” state, that is triggered when $v_{tank} = NaN$, and the transition to the “Fault” state, that is triggered by the values of the two other inputs of the model, i.e., p_s and ϕ_p. Specifically, p_s denotes the power supply to the station, while ϕ_p denotes a technical fault within the station. Thus, as shown in Fig. 17, if either the power supply is below the necessary power

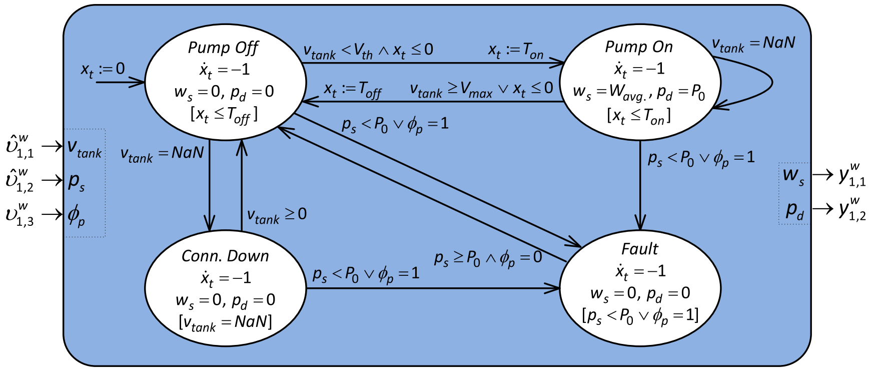

Fig. 17. Open hybrid automaton for the water pumping station.

demand of the station i.e., $p_s < P_0$, or when there is a technical fault i.e., $\phi_p = 1$, a transition to the "Fault" state will occur.

Appendix B. Telecommunications infrastructure components

B.1. Telecom central office

The telecom central office is modeled with the open hybrid automaton shown in Fig. 18. The model consist of five discrete states: (i) "Normal Ops." when the office carries its normal operations, (ii) "UPS usage" when the UPS system is in use due to a power cut, (iii) "Cooling Alarm" when the cooling system experiences problems due to water shortage, (iv) "Critical" when both (ii) and (iii) occur at the same time, and (v) "Fault" when the office becomes non-operational due to one or a combination of reasons such as, technical fault, depletion of the UPS batteries, and extended water shortage that shuts off the cooling system. As shown in Fig. 18, the transitions between the discrete states are triggered by the various guard conditions, that use the values of inputs and continuous states of the model. Specifically, there are three inputs, (i) p_s that denotes the power supply, (ii) w_s that denotes the water supply, and (iii) ϕ_T that denotes the existence of a technical fault. The model will transition from the "Normal Ops." to some other discrete state if there is a technical fault (i.e., $\phi_T = 1$) or if either the power supply p_s or water supply w_s is below the required power demand P_0 or water demand W_0, respectively. There are also two continuous states, x_{ups} and x_{wtr}, that are both used as timers. Specifically, the UPS system can keep the office operational after a power cut for T_{ups} time units and x_{ups} is used as a timer for that. Similarly, the cooling system can be functional after a water shortage for T_{wtr} time units and x_{wtr} is used as a timer for that. Finally, the three outputs of the model, z_{tco}, that denotes the status of the office, and p_d and w_d that denote the power demand and the water demand of the office respectively, change according to the discrete state of the model. For instance, $z_{tco} = 1$ in all states except the "Fault" state to denote that the office is operational, while $p_d = P_0$ and $w_d = W_0$ in the states where the office receives power and water, respectively.

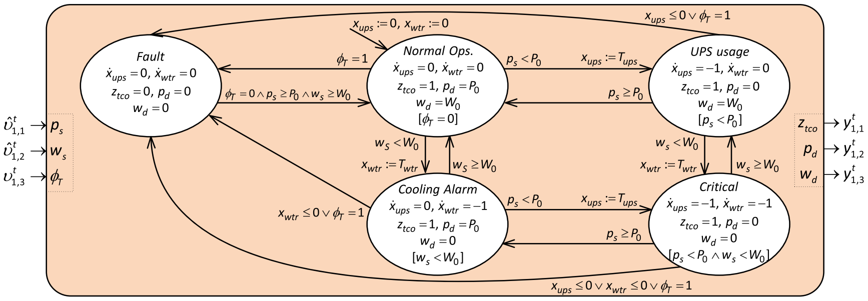

Fig. 18. Open hybrid automaton for the telecom central office.

B.2. Telecommunications network

The telecommunication network is modeled with the open hybrid automaton shown in Fig. 19, based on the single link model in [32]. The model consists of three discrete states: (i) "Steady-State", (ii) "Link Failure", and (iii) "Net Down". The model will remain at the "Steady-State" if there is no link failure, which we denote with the input $\phi_l = 0$, and also if the telecom central office operates normally, which we denote with the input $z_{tco} = 1$. While in "Steady-State", the transmission rate r (packets/sec) is calculated with $r = \frac{p}{T_p + q/B}$, where p is the number of packets entering the network, T_p is the propagation delay, and the quotient q/B is the queueing time with q being the aggregated queue size that is calculated by the continuous dynamics $\dot{q} = r - B$ (note that q is always ≥0) and B being the network bandwidth (packets/sec) [32]. Based on the transmission rate r, the three input signals, s_1 that denotes the SCADA's control signal to the substation, s_2 that denotes the tank's volume measurement, and s_3 that denotes the substation's power measurement, are delayed through the network according to $s_k(t) = s_k(t + \frac{P_k}{r})$, $k = 1, 2, 3$, where P_k denotes the number of packets that are sent for each signal k. Thus, the more traffic in the network, the larger the transmission delay. In case link(s) fail, i.e., $\phi_l > 0$, there will be a transition to the "Link Failure" state, where the bandwidth of the network is reduced following $B_\phi = B - a\phi_l$, where a associates the number of failed links ϕ_l with the bandwidth B. The new bandwidth B_ϕ is used to calculate r and q while in this state. Finally, in case the telecom central office fails, i.e., $z_{tco} = 0$, there will be a transition to the "Net Down" state, and, as a result, all the output signals will become $s_k = NaN$.

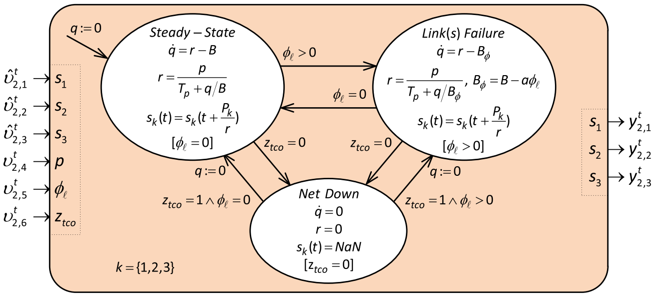

Fig. 19. Open hybrid automaton for the telecommunications network.

References

[1] Commission of the European Communities, European Programme for Critical Infrastructure Protection, Communication from the Commission on a European Programme for Critical Infrastructure Protection; 2006. URL ⟨http://goo.gl/iXWTT4⟩.

[2] Rinaldi S, Peerenboom J, Kelly T. Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Syst 2001;21(6):11–25. http://dx.doi.org/10.1109/37.969131.

[3] Ouyang M. Review on modeling and simulation of interdependent critical infrastructure systems. Reliab Eng Syst Saf 2014;121:43–60. http://dx.doi.org/10.1016/j.ress.2013.06.040.

[4] Wang S, Hong L, Ouyang M, Zhang J, Chen X. Vulnerability analysis of interdependent infrastructure systems under edge attack strategies. Saf Sci 2013;51(1):328–37. http://dx.doi.org/10.1016/j.ssci.2012.07.003.

[5] O'Rourke TD, Lembo AJ, Nozick LK. Lessons learned from the world trade center disaster about critical utility systems. In: Monday JL, (Ed.), Beyond September 11th: An Account of Post-Disaster Research, Natural Hazards Research and Applications Information Center, Boulder, CO; 2003. p. 269–90. URL ⟨http://goo.gl/Vmm8pZ⟩.

[6] Simpson DM, Lasley CB, Rockaway TD, Weigel T. Understanding critical infrastructure failure: examining the experience of Biloxi and Gulfport, Mississippi after Hurricane Katrina. Int J Crit Infrastruct 2010;6(3):246–76. http://dx.doi.org/10.1504/IJCIS.2010.033339.

[7] Tanaka S. Accident at the Fukushima Dai-ichi nuclear power stations of TEPCO - Outline & lessons learned. In: Proceedings of the Japan Academy, Series B, vol. 88, 9; 2012. p. 471–84. ⟨http://dx.doi.org/10.2183/pjab.88.471⟩.

[8] McDaniels T, Chang S, Peterson K, Mikawoz J, Reed D. Empirical framework for characterizing infrastructure failure interdependencies. J Infrastruct Syst 2007;13(3):175–84. http://dx.doi.org/10.0/bg8cd3.

[9] Rinaldi S. Modeling and simulating critical infrastructures and their interdependencies. In: Proceedings of the 37th annual Hawaii international conference on system sciences; 2004, p. 8. ⟨http://dx.doi.org/10.0/cwgt95⟩.

[10] Satumtira G, Dueñas-Osorio L. Synthesis of modeling and simulation methods on critical infrastructure interdependencies research. Springe Berl Heidelb 2010:1–51. http://dx.doi.org/10.1007/978-3-642-11405-2_1.

[11] Heracleous C, Panayiotou CG, Polycarpou MM, Ellinas G. Modeling interdependent critical infrastructures using open hybrid automata. In: Proceedings of IEEE conference on computer communications workshops (INFOCOM WKSHPS); 2015, p. 671–6. ⟨http://dx.doi.org/10.0/bm6b⟩.

[12] Hasan S, Foliente G. Modeling infrastructure system interdependencies and socioeconomic impacts of failure in extreme events: emerging R & D challenges. Nat Hazards 2015;78(3):2143–68. http://dx.doi.org/10.0/bwjb.

[13] Heracleous C, Oliva G, Setola R. System of Systems Modeling – State of Art, Project FACIES Deliverable Work Package 200; 2013. URL ⟨https://goo.gl/IQDyFU⟩.

[14] Pederson P, Dudenhoeffer D, Hartley S, Permann M. Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research. Tech. Rep., Idaho National Laboratory, Idaho Falls; 2006. URL ⟨https://goo.gl/9fRMpy⟩.

[15] Haimes YY, Jiang P. Leontief-based model of risk in complex interconnected infrastructures. J Infrastruct Syst 2001;7(1):1–12. http://dx.doi.org/10.0/cjkbmp.

[16] Oliva G, Panzieri S, Setola R. Fuzzy dynamic input-output inoperability model. Int J Crit Infrastruct Prot 2011;4(3–4):165–75. http://dx.doi.org/10.1016/j.ijcip.2011.09.003.

[17] Dudenhoeffer DD, Permann MR, Manic M. CIMS: A framework for infrastructure interdependency modeling and analysis. In: Proceedings of the 2006 winter simulation conference; 2006. p. 478–85. ⟨http://dx.doi.org/10.0/dmz7qk⟩.

[18] Kaegi M, Mock R, Kroger W. Analyzing maintenance strategies by agent-based simulations: a feasibility study. Reliab Eng Syst Saf 2009;94(9):1416–21. http://dx.doi.org/10.1016/j.ress.2009.02.002.

[19] Svendsen NK, Wolthusen SD. Graph models of critical infrastructure interdependencies. Heidelberg: Springer Berlin; 2007. p. 208–11. http://dx.doi.org/10.1007/978-3-540-72986-0_27.

[20] Hernandez-Fajardo I, Duenas-Osorio L. Probabilistic study of cascading failures in complex interdependent lifeline systems. Reliab Eng Syst Saf 2013;111:260–72. http://dx.doi.org/10.1016/j.ress.2012.10.012.

[21] Gursesli O, Desrochers AA. Modeling infrastructure interdependencies using Petri nets. In: Proceedings of IEEE international conference on systems, man and cybernetics, vol. 2; 2003, p. 1506–12. ⟨http://dx.doi.org/10.0/d6c9b8⟩.

[22] Beccuti M, Chiaradonna S, Giandomenico FD, Donatelli S, Dondossola G, Franceschinis G. Quantification of dependencies between electrical and information infrastructures. Int J Crit Infrastruct Prot 2012;5(1):14–27. http://dx.doi.org/10.0/fxmc8p.

[23] Di Giorgio A, Liberati F, Bayesian Network-Based A. Approach to the critical infrastructure interdependencies analysis. IEEE Syst J 2012;6(3):510–9. http://dx.doi.org/10.1109/JSYST.2012.2190695.

[24] Min H-SJ, Beyeler W, Brown T, Son YJ, Jones AT. Toward modeling and simulation of critical national infrastructure interdependencies. IIE Trans 2007;39(1):57–71. http://dx.doi.org/10.1080/07408170600940005.

[25] Porcellinis SD, Panzieri S, Setola R. Modelling critical infrastructure via a mixed holistic reductionistic approach. Int J Crit Infrastruct 2009;5(1–2):86–99. http://dx.doi.org/10.1504/IJCIS.2009.022851.

[26] Nan C, Eusgeld I. Adopting HLA standard for interdependency study. Reliability Engineering & System Safety, Special Issue on Safecomp 2008, 96 1; 2011. p. 149–59. ⟨http://dx.doi.org/10.1016/j.ress.2010.08.002⟩.

[27] Eusgeld I, Nan C, Dietz S. System-of-systems approach for interdependent critical infrastructures. Reliability Engineering & System Safety 96, 6; 2011. p. 679–86, ESREL 2009 Special Issue. ⟨http://dx.doi.org/10.0/dp9wkr⟩.

[28] Lygeros J, Pappas G, Sastry S. An introduction to hybrid system modeling, analysis, and control. Preprints of 1st Nonlinear Control Network Pedagogical School. URL ⟨http://goo.gl/uif2fI⟩.

[29] Hiskens I, Pai M. Hybrid systems view of power system modelling, In: Proceedings of the IEEE international symposium on circuits and systems. ISCAS 2000, Geneva. vol. 2, 2000, p. 228–31. ⟨http://dx.doi.org/10.1109/ISCAS.2000.856300⟩.

[30] Fourlas G, Kyriakopoulos K, Vournas C. Hybrid systems modeling for power systems. IEEE Circuits Syst Mag 2004;4(3):16–23. http://dx.doi.org/10.1109/MCAS.2004.1337806.

[31] Dhople S, Chen Y, DeVille L, Dominguez-Garcia A. Analysis of power system dynamics subject to stochastic power injections. IEEE Trans Circuits Syst I: Regul Pap 2013;60(12):3341–53. http://dx.doi.org/10.1109/TCSI.2013.2265972.

[32] Lee J, Bohacek S, Hespanha JP, Obraczka K. Modeling communication networks with hybrid systems. IEEE/ACM Trans Netw 2007;15(3):630–43. http://dx.doi.org/10.1109/TNET.2007.893090.

[33] Mihaylova L, Boel R. Hybrid stochastic framework for freeway traffic flow modelling. In: Proceedings of the 1st international symposium on information and communication technologies; 2003, p. 372–7. URL ⟨http://doi.acm.org/10.1145/963600.963674⟩.

[34] Noori A, Menhaj MB, Shafiee M. Hybrid systems modeling for gas transmission network. In: Proceedings of the 4th IFAC conference on management and control of production and logistics. Volumes 40, 18; 2007. p. 389–96 ⟨http://dx.doi.org/10.3182/20070927-4-RO-3905.00065⟩.

[35] Lemch ES, Guay M, Rudie K. Modelling of oil production operation in the framework of hybrid systems. In: Proceedings of the 41st IEEE conference on decision and control, vol. 2; 2002, p. 1595–1600. ⟨http://dx.doi.org/10.1109/CDC.2002.1184747⟩.

[36] Johansson J, Hassel H. An approach for modelling interdependent infrastructures in the context of vulnerability analysis. In: Proceedings of the 19th European safety and reliability conference on reliability engineering & system safety. 95; 12; 2010. 1335–44 ⟨http://dx.doi.org/10.1016/j.ress.2010.06.010⟩.

[37] Caughlin D, Sisti AF. Summary of model abstraction techniques. In: Proceedings SPIE 3083, Enabling Technology for Simulation Science, 1997. ⟨http://dx.doi.org/10.1117/12.276726⟩.

[38] Zeigler BP. Review of theory in model abstraction. In: Proceedings SPIE 3369, Enabling Technology for Simulation Science II, 1998. ⟨http://dx.doi.org/10.0/c6hm5p⟩.

[39] Davis PK. An introduction to variable-resolution modeling. Nav Res Logist (NRL) 1995;42(2):151–81. http://dx.doi.org/10.0/cgkq4r.

[40] Nan C, Sansavini G. Multilayer hybrid modeling framework for the performance assessment of interdependent critical infrastructures. Int J Crit Infrastruct Prot 2015;10:18–33. http://dx.doi.org/10.1016/j.ijcip.2015.04.003.

[41] Liu J, Liu X, Koo T-KJ, Sinopoli B, Sastry S, Lee E. A hierarchical hybrid system model and its simulation. In: Proceedings of the 38th IEEE conference on decision and control, vol. 4; 1999. p. 3508–13. ⟨http://dx.doi.org/10.1109/CDC.1999.827883⟩.

[42] Lygeros J, Tomlin C, Sastry S. Hybrid Systems: Modeling, Analysis and Control; 2008. URL ⟨http://goo.gl/KT2IvV⟩.

[43] Lynch N, Segala R, Vaandrager F. Hybrid I/O Automata Revisited. Vol. 2034 of Lecture Notes in Computer Science; 2001, p. 403–17. ⟨http://dx.doi.org/10.1007/3-540-45351-2_33.

[44] Bornot S, Sifakis J. On the composition of hybrid systems. In: Hybrid Systems: Computation and Control, Vol. 1386 of Lecture Notes in Computer Science, 1998, pp. 49–63. ⟨http://dx.doi.org/10.1007/3-540-64358-3_31⟩.

[45] Sahbani A, Pascal J. Simulation of hyibrd systems using Stateflow. In: Proceedings of European simulation multiconference on simulation and modelling (ESM); 2000, p. 271–5. URL ⟨http://goo.gl/8XvAKm⟩.