Chapter 17

AN INTEGRATED APPROACH FOR SIMULATING INTERDEPENDENCIES

Roberto Setola, Sandro Bologna, Emiliano Casalicchio and Vincenzo Masucci

Abstract The detailed simulation of interdependent critical infrastructures is a hard problem. Major challenges include modeling multiple heterogeneous infrastructures in a single framework and expressing internal dependencies and interdependencies between infrastructures. This paper attempts to address these issues by proposing a simulation framework where several sector-specific simulators (vertical simulators) are integrated into a general simulation environment (horizontal simulator). Specialized software implemented in the vertical simulators models individual infrastructures and their intra-domain dynamics. The horizontal simulator effectively captures inter-domain relationships and merges heterogeneous information from the vertical simulators to facilitate comprehensive infrastructure simulations.

Keywords: Heterogeneous infrastructures, interdependencies, simulation

1. Introduction

Predicting the behavior of critical infrastructures during crisis or failure conditions is a difficult task. Innovative models and tools are needed to capture and reason about the behavior of critical infrastructures. However, the complexity of the problem severely limits the application of analytic methods. Consequently, simulation methods are increasingly used to investigate the behavior of interdependent infrastructures (see, e.g., [3, 15]).

The primary challenge in critical infrastructure simulation is to model heterogeneous behavior in a single framework. Another major challenge is to model interdependencies between infrastructures and interdependencies between infrastructures and the external environment.

Simulation techniques may be broadly divided into two classes. The first uses domain-specific simulators, each designed for detailed simulations of a single infrastructure. The second engages newer simulation environments where the behavior of different infrastructures is modeled using a single conceptual framework.

We refer to the first class as federated simulators because several "vertical simulators" exchange data. A vertical simulator considers an infrastructure as an autonomous, isolated system; hence, it provides a vertical (partial but detailed) view of the infrastructure. A good example of such a federated simulator is EPOCHS [10], which is designed to analyze interactions between the electric grid and telecommunications networks. Another example is SimCIP, which is being developed under the IRRIIS Project [11].

This class of simulators leverages well-tested simulation packages and data and models accumulated over years of use, which reduces development costs. Moreover, because each simulator is designed for a specific domain, the knowledge elicitation phase is greatly simplified: the simulator and the user engage the same language and vocabulary (i.e., modeling schemas and paradigms). Furthermore, all the intra-domain dependencies are modeled within vertical simulators; the simulation environment only manages the inter-domain dependencies.

Unfortunately, these simulators are unable to capture all the distinct elements that characterize real-world scenarios. In many cases, they describe infrastructure interactions only in terms of functional links (e.g., direct exchanges of goods and services), neglecting other types of interdependencies (e.g., geographical or social links) [16]. Also, because vertical simulators cannot manage information belonging to other domains (except for possibly changing loads and certain model parameters), they are able to reproduce only very elementary interaction mechanisms.

The second class of simulation approaches uses a sufficiently broad and powerful modeling framework to represent multiple heterogeneous infrastructures [1, 5, 8, 14, 17]. We refer to this type of simulator as a "horizontal simulator" because it covers multiple domains. Horizontal simulators engage a variety of concepts, structures and solutions, e.g., complex adaptive systems [16], agent-based modeling [2, 4] and entity-relation approaches [5]. These paradigms capture different aspects of the problem to greater or lesser degrees; therefore, the quality of their results vary according to the specific scenarios being evaluated.

Except for NISAC [17] and a few other government initiatives, horizontal simulators have been tested only on relatively simple scenarios. This is primarily due to modeling issues. It is extremely difficult to acquire detailed information about infrastructure parameters and the dynamical behavior of infrastructures. Part of the problem is that critical infrastructure stakeholders are reluctant to release data that they deem to be sensitive. Also, it is difficult to translate information from infrastructure domains to the abstract formulation adopted by horizontal simulators.

To address the limitations of the two classes of simulation approaches, we propose a simulation framework that integrates multiple vertical simulators in a general horizontal simulation environment. The architecture, which is presented in Figure 1, has three layers: (i) a user layer where different scenarios

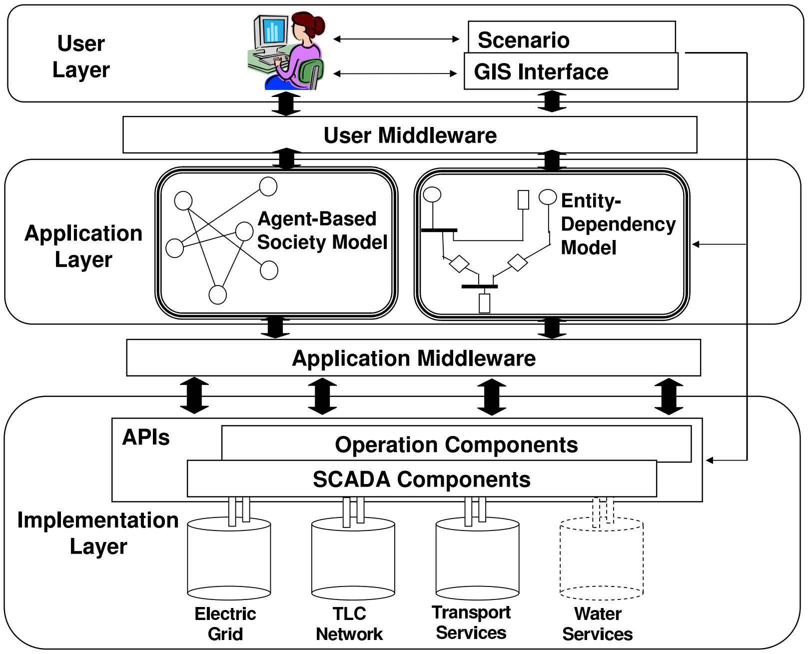

Figure 1. Simulation framework.

are described in terms of component elements, links between elements, event sequences, and simulation tools to be used; (ii) an application layer that manages a collection of horizontal simulators (CISIA [5] and FederatedABMS [4]); and an implementation layer that manages a set of vertical simulators (OMNeT++ [13], ns2 [9] and e-Agora). Specifically, once a scenario is selected, the user can choose the type of horizontal simulator to be used. Moreover, the user can choose to reproduce everything within the selected simulator or to delegate part of the simulation effort (e.g., intra-domain dynamics) to vertical simulators in the implementation layer.

The simulation framework is being developed under the CRESCO Project [12], which is funded by Italian Ministry of Research to enhance the national capability to study complex systems. However, we must emphasize that this framework is not intended to be a "silver bullet" that comprehensively addresses the critical infrastructure simulation problem. Rather, it is a proof-of-concept scheme that, we hope, will constitute the first step to creating a European Infrastructure Simulation and Analysis Center (EISAC).

2. CRESCO Architecture

The CRESCO architecture is characterized by the presence of horizontal simulation environments that are used to integrate a set of vertical simulators.

We represent the status of the i-th component of the x-infrastructure as x_i, where the status refers to any quantity that describes a characteristic (efficiency, operability, etc.) of the element. In general, an element is described by several variables, $x_i \in \Re^{n_i}$ where $n_i \geq 1$.

A vertical simulator updates the status of elements by solving an equation of the form:

$$x_i(k+1) = f_X(X(k), u(k), \bar{P}_{x_i}) \tag{1}$$

where $X = [x_1, \ldots, x_n]^T$ represents the status of all the elements comprising the x-infrastructure, u is the vector of external inputs (e.g., loads), $\bar{P}_{x_i}$ is the set of parameters characterizing the i-th element and $f_X(\cdot)$ is a function that describes the dynamics of the x-infrastructure. The form of the function depends on the nature of the infrastructure and may be expressed via differential equations, algebraic relations, etc. Note that $f_X(\cdot)$ also takes into account intra-domain relationships existing between elements in the x-infrastructure.

If two or more vertical simulators are federated using a simple broker gateway, Equation 1 is modified to account for the presence of interdependencies. For simplicity, we consider two infrastructures, x and y, and assume that a one-to-one correspondence exists between elements in the two infrastructures (this assumption can be relaxed). In this case, Equation 1 becomes

$$x_i(k+1) = f_X(X(k), u(k) + \delta(y_i), \gamma(\bar{P}_{x_i}, y_i)) \tag{2}$$

where δ and γ are functions that map the values assumed by the state variable of the y-infrastructure to load variations or parameter changes for the x-infrastructure element. Hence, the federation process corresponds to a fictitious modification of the load u and/or parameters $\bar{P}_{x_i}$, making them dependent on the status assumed by the element of the y-infrastructure that is in direct correspondence with x_i.

As mentioned above, this strategy has several drawbacks. In particular, it only permits the reproduction of simple interdependency phenomena.

The dynamics of an element in a horizontal simulator is described by:

$$\theta_i(k+1) = \Gamma_\theta(\theta_i(k), u(k), M(\Theta)) \tag{3}$$

where $\theta_i \in \Re^{n_i}$ is the state of the i-th element, Θ is the vector of state variables of elements in the simulation scenario, Γ_θ is a function describing the dynamics of each element and $M(\cdot)$ is a function describing the inter- and intra-domain relationships existing between the i-th element and all the other elements without any specific consideration about the nature of the infrastructure. Note that the horizontal simulation techniques proposed in the literature differ substantially in the methodology used to map interdependencies to the $M(\cdot)$ function and in the formalism adopted for their representation.

The main difference between Equations 1 and 3 is that $f_X(\cdot)$ depends explicitly on all the variables of the system while Γ_θ depends only on the state variables of the i-th element; this is because the influence exerted by the other

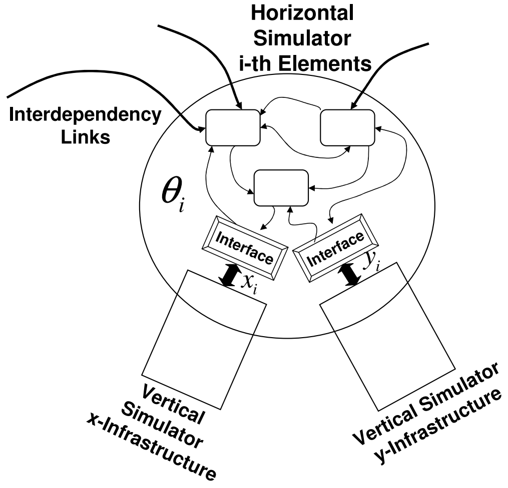

Figure 2. Global behavior of the i-th element.

elements is mediated by $M(\cdot)$, which codifies the relationships between the different elements. Therefore, Equation 3 provides a less detailed and less coherent representation of intra-domain relationships than Equation 1. However, it is able to describe inter-domain dependencies better because a more general formalism is adopted.

We believe that some of the drawbacks can be overcome by integrating several vertical simulators via a horizontal simulation framework. This strategy leverages the capabilities of the vertical simulators that correctly reproduce the behavior of individual infrastructures and the ability of a horizontal simulator to model a large class of inter-domain relationships.

Figure 2 illustrates how the global behavior of the i-th element is obtained by using Equation 3 to model inter-domain relationships along with information provided by a set of vertical simulators. Specifically, the overall state of the i-th element (θ_i) is obtained by incorporating, within the horizontal simulator model, information about the "partial" view of the element obtained by considering the x- and y-infrastructure models (x_i and y_i, respectively). Moreover, θ_i is used to update the variables of the vertical simulators in order to propagate the consequences of inter-domain phenomena in those environments. Formally, we have:

$$\begin{aligned}
\theta_i(k+1) &= \Gamma_\theta(\tilde{\theta}_i(k), u(k) + \delta_\theta u\,(x_i, y_i)\,, \hat{M}(\Theta)) \\
x_i(k+1) &= f_X(\tilde{X}(k), u(k) + \delta u_x(\theta_i), \gamma(\bar{P}_{x_i}, \theta_i)) \\
y_i(k+1) &= f_Y(\tilde{Y}(k), u(k) + \delta u_y(\theta_i), \gamma(\bar{P}_{y_i}, \theta_i))
\end{aligned} \tag{4}$$

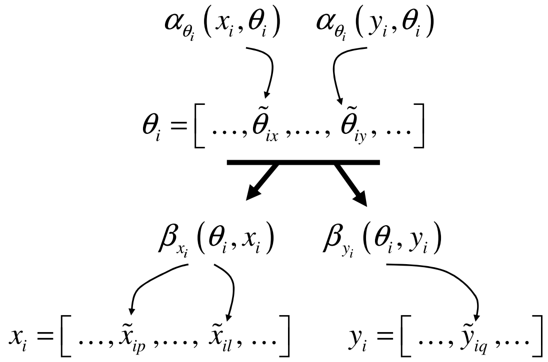

Figure 3. Mappings between vertical and horizontal simulator variables.

where $\hat{M}(\cdot)$ is a function that considers only inter-domain relationships and the $\tilde{}$ operator means that the corresponding variable has been updated with regard to its own (i.e., isolated) value based on data provided by other simulators.

Figure 3 shows how the interface components in Figure 1 implement ontology mappings. Note that the functions $\alpha_{\theta_i}(x_i, \theta_i)$ and $\beta_{x_i}(\theta_i, x_i)$ translate vertical simulator quantities into horizontal simulator variables and vice versa.

At each iteration, the horizontal simulator evaluates the overall status of each component in the modeled scenario. This information is translated into quantities recognized by the vertical simulators and input to them. Then, the vertical simulators, using detailed models of each infrastructure, update the network configuration in terms of the actual loads/resources, availability of components, etc. Finally this data, upon being codified appropriately, is sent back to the horizontal simulator, where it is used to refine the status of the element.

Figure 4 presents a simulation scenario. The horizontal simulator has four components: Generation Plant (A), Control Center (B), Urban Area (C) and Distribution Substation (D). Some of these elements have a complex structure, i.e., their behavior is described by considering the results of the union of several heterogeneous aspects. For example, in order to analyze the Generation Plant, it is necessary to consider its electrical behavior (provided by the Electric Grid Simulator) and status of its telecommunications systems (provided by the Telecommunications Network Simulator). However, these are only two "partial views" of the element. To obtain the overall status, it is necessary to integrate them with information related to internal dynamics and the effects induced on the element by inter-domain relationships. This task is performed by the horizontal simulator. Obviously, the overall status of the Generation Plant influences both the Electric Grid and Telecommunications Network. Hence, this status should be made available to the vertical simulators that propagate the effects in their domains.

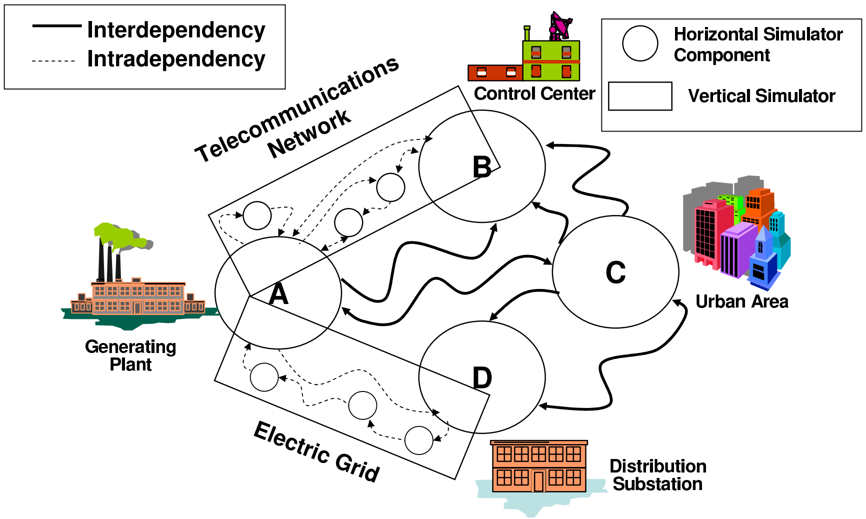

Figure 4. Two vertical simulators integrated in a horizontal simulation environment.

3. Simulation Framework

The CRESCO architecture provides facilities for defining and configuring simulation scenarios, a model for analyzing critical infrastructure interdependencies, and middleware that allows sector-specific models to be integrated in order to simulate the detailed behavior of critical infrastructures. The user layer resides at the top level of the three-layer architecture (Figure 1). It provides an interface for defining and configuring simulation scenarios, entering simulation parameters, and manipulating and visualizing simulation results. Two approaches are considered, CISIA based on the Entity Resource Model [5] and FederatedABMS, which employs an agent-based model [2, 4]. The lowest implementation layer incorporates vertical models of sector-specific infrastructures that are implemented by vertical simulators. These simulators are instantiated at runtime on one or more nodes of the ENEA GRID.

The vertical simulators currently employed in the CRESCO architecture are designed to simulate telecommunications networks, computer systems and the electric power grid. Open-source simulation frameworks that can be modified to work in the CRESCO environment have been adopted to the extent possible. OMNeT++ [13] and ns2 [9] are used for telecommunications network and computer systems modeling while e-AGORA [1], a load flow simulation environment, is used to simulate electricity distribution networks and the power grid.

Application middleware supports interoperability between horizontal and vertical simulators (Figure 1). The middleware coordinates mappings between elements of different simulators. This is facilitated by a common ontology for data and event exchange between different simulation frameworks (despite their disparate representations). The events produced by the horizontal simulators are mapped via wrappers (an event transcoding plug-in for each interface) to a set of standard events. These quantities are then converted by the appropriate transcodifying ontologies module to a format that is compatible with the specific vertical simulator.

4. Simulator Integration

The most innovative element of the CRESCO architecture is the presence of horizontal and vertical simulators in a single framework. This section describes how horizontal simulators in the application layer are integrated with vertical simulators in the implementation layer. For reasons of space, we only describe the integration of the CISIA horizontal simulator.

4.1 Rationale

The decision to incorporate multiple horizontal simulators was motivated by the need to demonstrate a proof-of-concept architecture that is both flexible and scalable. Another, more important reason is the ability to correctly model various interdependencies and their phenomena. Indeed, as mentioned earlier, existing models do not capture interdependencies, which severely limits the fidelity of their simulations. Also, providing users with multiple tools to investigate scenarios, enables them to conduct better analyses.

FederatedABMS and CISIA adopt similar conceptual representations; this makes the scenarios interchangeable and the results comparable. However, the two horizontal simulators have several different and complementary characteristics. FederatedABMS is an event-driven simulator while CISIA is a discrete-time simulator. FederatedABMS adopts an agent-based formulation; thus, the function $M(\cdot)$ in Equation 3 has the form:

$$M(\Theta) \cong m(k)\Theta. \tag{5}$$

Since relations between agents can be created or destroyed during a simulation, a time-varying incident matrix is employed to codify these relationships.

On the other hand, CISIA is based on the concept of proximity and employs the formulation:

$$M(\Theta) \cong \sum_{l} M_l(\Theta) \tag{6}$$

where each element $M_l(\cdot)$ in the summation represents a specific mechanism of interrelation. Obviously, neither Equation 5 nor 6 is exhaustive and the implementations of the two simulators impose further limitations. Nevertheless, they provide two interesting and, in certain respects, complementary views of interdependencies. Indeed, the concurrent use of FederatedABMS and CISIA helps users better understand the phenomena produced by interdependencies.

4.2 CISIA Integration

CISIA is a simulation environment designed to analyze complex scenarios involving multiple heterogeneous infrastructures with tight interactions. CISIA's abstract representation decomposes a scenario into macro-components whose behavior is described in terms of their ability to produce goods and services based on the availability of external resources while taking into account the presence (and severity) of failures. Each entity computes its "operative level" (i.e., ability to perform its intended job) and the level of severity associated with different types of failures based on the internal state and external resources; the presence and the severity of internal and external failures are also considered. These quantities are then exchanged among the macro-components to estimate the overall behavior of the system. CISIA's representation employs fuzzy numbers [5, 14] to accommodate uncertainty and capture the linguistic descriptions of system states provided by human experts.

CISIA adopts a resource-service paradigm to facilitate the integration of vertical simulators. Thus, the outputs (x_i and y_i in Equation 4) can be logically managed as external resources. However, it is necessary to translate these quantities to conform with CISIA's more abstract representation. This is accomplished by aggregating data using the appropriate ontology.

The implementation of information flow from CISIA to the vertical simulators is a challenging problem. This is because the software packages were not designed to be integrated with other simulation environments. Moreover, abstract information that characterizes states in CISIA has to be mapped to tangible quantities such as electric power supply parameters. This process involves data decomposition, which introduces a degree of arbitrariness. The process is also complicated by the fact that CISIA codifies data in the form fuzzy numbers. While fuzzy numbers facilitate the fusion of data provided by vertical simulators, they render the inverse process almost intractable. Indeed, in such an instance, it is necessary to convert fuzzy numbers to crisp values, but this leads to information loss.

Another important issue is time synchronization. Since CISIA is a discrete-time simulator, the implementation middleware has to operate as a scheduler to activate the different simulators. Specifically, CISIA performs a simulation for one "time tick"(e.g., one minute in a simulation), then the middleware supplies the vertical simulators with the output of CISIA and activates their execution. When the vertical simulators have completed their calculations, the outputs are sent to CISIA, which starts the simulation for next time tick. Note that when a discrete-time vertical simulator (e.g., ns2) is used, the scenario has to be analyzed in an interrelated manner, which imposes additional constraints on the time tick period.

5. Conclusions

The CRESCO architecture attempts to address the limitations of existing critical infrastructure simulation methods by integrating multiple vertical simulators in a general horizontal simulation environment. The vertical simulators effectively model individual infrastructures and intra-domain dynamics while the horizontal simulators express inter-domain relationships and merge heterogeneous information from different infrastructures. The architecture is intended to be a proof of concept, the first step in creating a comprehensive simulation tool for analyzing multiple critical infrastructures that are tightly coupled and highly interdependent.

Our future research will attempt to develop an ontology that formalizes information flow between vertical simulators and the horizontal simulation environment. The IRRIIS Project [11] has taken a step in this direction by defining an information flow model that supports interdependency analysis; a more formal approach is being investigated by the DIESIS Project [6]. In addition, we will investigate the granularity of domain models used in vertical simulators. These results will assist in integrating information from multiple heterogeneous simulators during large-scale critical infrastructure simulations.

Acknowledgements

The authors wish to acknowledge the technical assistance provided by members of the CRESCO Project: E. Ciancamerla, S. De Porcellinis, S. Di Blasi, G. Di Poppa, E. Galli, A. Jannace, E. Marchei, M. Minichino, S. Podda, S. Ruzzante, A. Tofani and G. Vicoli.

References

[1] C. Balducelli, S. Bologna, L. Lavalle and G. Vicoli, Safeguarding information intensive critical infrastructures against novel types of emerging failures, *Reliability Engineering and System Safety*, vol. 92(9), pp. 1218–1229, 2007.

[2] V. Cardellini, E. Casalicchio and E. Galli, Agent-based modeling of interdependencies in critical infrastructures through UML, *Proceedings of the Agent-Directed Simulation Symposium of the Spring Simulation Multiconference* (www.ce.uniroma2.it/publications/ads2007.pdf), 2007.

[3] E. Casalicchio, P. Donzelli, R. Setola and S. Tucci, Modeling and simulation of interdependent critical infrastructures: The road ahead, in *Communication Networks and Computer Systems: A Tribute to Professor Erol Gelenbe*, J. Barria (Ed.), Imperial College Press, London, United Kingdom, pp. 143–157, 2006.

[4] E. Casalicchio, E. Galli and S. Tucci, Federated agent-based modeling and simulation approach to study interdependencies in IT critical infrastructures, *Proceedings of the Eleventh International Symposium on Distributed Simulation and Real-Time Applications*, pp. 182–189, 2007.

[5] S. De Porcellinis, R. Setola, S. Panzieri and G. Ulivi, Simulation of heterogeneous and interdependent critical infrastructures, *International Journal of Critical Infrastructures*, vol. 4(1/2), pp. 110–128, 2008.

[6] DIESIS Consortium, The DIESIS Project (www.diesis-eu.org).

[7] D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1998.

[8] D. Dudenhoeffer, M. Permann and M. Manic, CIMS: A framework for infrastructure interdependency modeling and analysis, *Proceedings of the Winter Simulation Conference*, pp. 478–485, 2006.

[9] K. Fall and K. Varadhan, The Network Simulator – ns-2 (nsnam.isi.edu/nsnam/index.php/Main_Page).

[10] K. Hopkinson, R. Giovanini, X. Wang, EPOCHS: Integrated commercial off-the-shelf software for agent-based electric power and communication simulation, *Proceedings of the Winter Simulation Conference*, pp. 1158–1166, 2003.

[11] IRRIIS Consortium, The IRRIIS European Integrated Project, Fraunhofer Institute for Intelligent Analysis and Information Systems, Sankt-Augustin, Germany (www.irriis.org).

[12] Italian National Agency for New Technologies, Energy and the Environment (ENEA), The CRESCO Project, Rome, Italy (www.cresco.enea.it).

[13] OMNeT++ Community Site, OMNeT++ Discrete Event Simulation System (www.omnetpp.org).

[14] S. Panzieri, R. Setola and G. Ulivi, An approach to model complex interdependent infrastructures, presented at the *Sixteenth IFAC World Congress* (www.dia.uniroma3.it/∼panzieri/Articoli/WorldIFAC05-CIIP.pdf), 2005.

[15] P. Pederson, D. Dudenhoeffer, S. Hartley and M. Permann, Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research, Report No. INL/EXT-06-11464, Critical Infrastructure Protection Division, Idaho National Laboratory, Idaho Falls, Idaho, 2006.

[16] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies, *IEEE Control Systems*, vol. 21(6), pp. 11–25, 2001.

[17] Sandia National Laboratories, National Infrastructure Simulation and Analysis Center, Albuquerque, New Mexico (www.sandia.gov/mission/homeland/programs/critical/nisac.html).