An Integrated Modelling Approach for Spatial-aware Federated Simulation

Jingquan Xie
Fraunhofer IAIS
Sankt Augustin, Germany
E-Mail: jingquan.xie@iais.fraunhofer.de

Rafal Kozik
University of Science and Technology (UTP), Bydgoszcz, Poland
E-Mail: rkozik@utp.edu.pl

Nikolas Flourentzou
University of Cyprus
Nicosia, Cyprus
E-Mail: flourentzou.nikolas@ucy.ac.cy

***Abstract*—Federated simulation provides a powerful means to analyze dynamics of inter-connected large-scale systems like networks of critical infrastructures (CI) including power supply systems, railway systems, telecommunication networks, etc. Building a sound federation model that can be used for federated simulations is however a challenging task, especially when the spatial aspects of the sophisticated networks need to be considered. In this paper, an integrated modelling approach is proposed to reduce the effort in building such spatial models for federated simulation. It aims to automatically produce consistent models that can be accepted by domain-specific simulators like SIEMENS PSS SINCAL and ns-3, in particularly for use cases where federation models evolve frequently. This is a work in progress. The modelling approach itself is general purpose and not limited to certain domains like power or telecommunication systems.**

Keywords-Federated Simulation; Spatial Information System; Integrated Modelling; Dependency Modelling; ns-3; SINCAL; Crisis Management; Training Application; Critical Infrastructure;

I. INTRODUCTION

Large-scale inter-connected systems like networks of critical infrastructures play a central role in our daily life in modern society. It is critical to be able to analyse their behaviours under both normal and stressed situations. Modern simulation technology provides a powerful means to achieve this by combining different domain-specific simulators together - federated simulation [1]. Each simulator is developed by experts in their domain like the electrical energy distribution or IP-based telecommunication systems. A simulation middleware, like the DIESIS [2], is then used to facilitate the federation among the simulators.

One of the challenging tasks in developing and deploying systems based on the federated simulation is building the consistent federation models, especially when the spatial aspects of network elements need to be considered. The root of this challenge lays on the domain-specific simulators, which are in most cases developed independently by different organisations with limited support for modelling spatial information. Moreover, each simulator has its own syntax to describe the models, for example, SINCAL [3] uses relational databases to store the model in a sophisticated schema while ns-3 [4] models can be hard-coded in the source code. The motivation of this work is trying to establish an integrated modelling approach through a unified spatial-aware interface for model developers of different domains. Models that can be accepted by domain-specific simulators can be automatically generated to keep the federation model on a high-level of consistency.

The related research work dealing with federated simulation involves two categories: model generation and runtime environment. Most of the work focuses on the runtime environment to provide an interoperable execution environment for simulator federation. High-Level Architecture (HLA) [5] is a standard for integrating different simulators and it is widely used in the military. The DIESIS federation middleware [2] developed in the EU-funded research project DIESIS focuses on the integration of critical infrastructure simulators with knowledge base support. I2Sim [6] provides a federation mechanism where a common model for all involved domains are needed. To our best knowledge, these systems do not provide sufficient support in spatial-aware dependency modelling out-of-box.

This paper is structured as follows: Section I provides introductory material and motivation of the proposed approach. It is followed by Section II that presents the requirements of the proposed approach for developing spatial-aware federated simulation models. Section III describes the architecture and interfaces of the integrated modelling environment. This is still work-in-progress and the preliminary results are presented in Section IV to help the reader further understand the benefits of the proposed approach. Finally, conclusions are given in Section V.

II. SPATIAL-AWARE FEDERATION MODELLING

In real world, most of the elements in the critical infrastructure networks locate somewhere (e.g., an urban area substation, an antenna for mobile communication). In a well-built model of these systems, they are Geo-referenced - for example, a transformer in the SINCAL [3] model can be assigned with a coordinate in the form of latitude and longitude. Similarly Internet-capable routers used for IP-based communication in ns-3 [4] models also possess spatial information. To our knowledge, domain-specific simulators like SINCAL per se do not have sufficient support for modelling these kinds of geographical information. The reason is that spatial attributes do not affect the simulation results and therefore is irrelevant for most of the simulation tasks. In federated simulation however, spatial information can be used to determine the dependencies between different domain models. For instance, if a cabinet is near a router, it is very likely that the router is powered by the electrical power from that cabinet. The rest of this section discusses the requirements that are essential for developing spatial-aware federation models.

A. Domain Elements as First-class Citizen

In traditional Geographical Information Systems (GIS), primitive spatial features include points, lines and polygons. They are first-class citizens in those systems. Most operations are designed in mind to handle these primitive objects.

For an integrated spatial modelling environment covering federated simulation, the concept is different. First-class citizen objects that need to be managed should be high-level, which again build on top of the traditional primitive objects like points and lines. For instance, instead of dealing with lines directly, the system should provide utilities to manage power lines with additional attributes like power voltage, resistance, etc. For each domain element, a set of attributes should be provided for model developers to manage different physical parameters associated with that element. As an example, a list of elements from the power domain is provided in Table I with the corresponding attributes. It is essential to identify all relevant physical parameters needed by the domain-specific simulators. Model developers are responsible for giving appropriate values during the modelling phase. By design, the provider of the parameters of a given domain is the simulator adapter, which will be elaborated in Section III.B.

TABLE I. THE PHYSICAL PARAMETERS FOR THE POWER SYSTEM ELEMENTS.

	Generation	*Lines*	*Loads*	*Transformer*	*Converter*
ID	Yes	Yes	Yes	Yes	Yes
Category	1	2	3	4	5
Volt rating	Yes	Yes	Yes	Yes × 2	Yes
Resistance	No	Yes	No	Yes	No
Losses	No	Yes	No	Yes	Yes
Dimension	3D	Length	3D	3D	3D
Network	HV	HV, MV, LV	MV, LV	HV, MV	HV, MV
Other	Cooling	Overvoltage	No	Overpower	Cooling

B. Dependency Modelling

Dependency is an essential part of the network of critical infrastructures. It is however a challenging task to model dependencies in federated simulation systems. The reason is twofold:

- Identifying these dependencies between different domain elements is difficult if not impossible. For instance, in a network of CI, a router produces communication services for SCADA systems that remotely control a secondary substation in the power distribution network. In most cases, it is not clear for model developers which router really provides the service due to confidential levels of these kinds of information for CI operators.
- Missing an established methodology with sufficient tool support to facilitate the model developers to map the real dependency into the federation model. During the past years, various approaches have already been proposed to model dependencies [6]-[9]; however, none of them provides enough tool support to really enable the dependency modelling. Furthermore, spatial support is completely missing in these approaches.

In the proposed approach, this issue is addressed by a novel yet pragmatic solution. Dependent domain elements in the federation model will be identified by the model developers. Based on this information, the generated model (see Section II.C) for each domain-specific simulator will contain an instance of the same dependent elements (e.g., if a router provides Internet service for SCADA systems). This router instance will be modelled twice: one in the telecommunication (communication provider) and the other in the electrical distribution model (power energy consumer). Since the targeted domain model is generated automatically and the generation process is automatic, this approach provides a scalable solution for large federation models.

C. Domain Model Generation

Domain-specific simulators are normally developed by different organisations in parallel. Each simulator, both proprietary and open-source, normally provides its own user interface for users to develop models. Subsequently, in most cases, the modelling results are persistent in a proprietary format that can only be correctly handled by the simulator itself. There are some efforts to define standardised ways to represent models for specific domains, like the Common Information Model (CIM) [10] model for power networks. However, we are not aware of any standard for interdomain model representation. To our best knowledge, there is not a unified model representation for multiple domains that can be directly accepted by different domain-specific simulators.

To handle this issue in a pragmatic way, the proposed approach exports separate federation models in a representation that can be accepted by domain-specific simulators directly. For that purpose, simulator adapters (for each domain-specific simulator) are included in the proposed approach. These adapters will be part of the software tool - the integrated modelling environment. The software interfaces between the modelling core and the adapters will be specified and published, so that third parties can also contribute their adapters, e.g., for coupling their own simulators into the modelling system, more details see Section III.B. This feature is of utmost importance to keep the consistency of frequently changing federation models.

III. INTEGRATED FEDERATION MODELLING ENVIRONMENT

The software tool that provides the features mentioned in Section II is called Integrated Federation Modelling Environment (FEMI). In the rest of this section, the system architecture, the software interface specification and the spatial support will be elaborated.

A. Architecture

FEMI is an HTML5 Web application with sophisticated spatial support. The reason to develop FEMI as a Web application instead of traditional Desktop application is twofold:

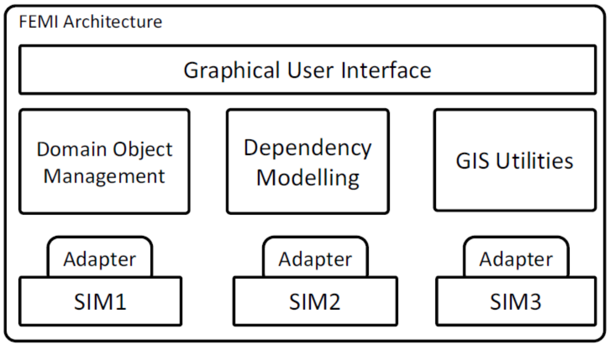

Figure 1. FEMI architecture.

- Internet provides the most pervasive infrastructure that can be accessed by different users from almost any locations. Browser-based applications that do not require an installation on the native machine greatly reduce the efforts needed to start working with the system.
- Tool support for developing HTML5 applications is evolving extremely fast due to the promotion of large IT companies. Frameworks and libraries like NodeJS [11] and LeafletJS [12] make the development of HTML5 Web applications with spatial support much easier than developing a Desktop application with comparable functionalities.

In general FEMI has three functional layers (see Fig. 1: 1) the Graphical User Interface (GUI) providing easy-to-use interfaces for model developers. FEMI GUI hides all of the low-level technical details of management internal data structures. 2) the management layer that provides support for domain object management, dependency modelling in federated simulations, and the GIS utilities. The domain object management is responsible for handling first-class citizens in FEMI like the power lines, the router, etc. The dependency modelling module is used to manage dependent elements. Finally, the GIS utilities provide sophisticated support for managing primitive spatial features like points and polygons. 3) the simulator layer contains various domain simulators and the adapters. The adapters work like a translator between the FEMI core system and the simulators, which per se do not understand any commands sent by FEMI.

B. Interface Specification

To enable the communication between FEMI core and the domain-specific simulators, a set of software interfaces are needed. On the functional level, these interfaces are the basis for the following:

- Expose the capabilities of the simulators. Capabilities include for instance if the simulator supports stepping functions during the simulation; what kind of physical parameters are needed for a given model element, etc. This interface is normally used during the initial phase before interacting with the simulators.

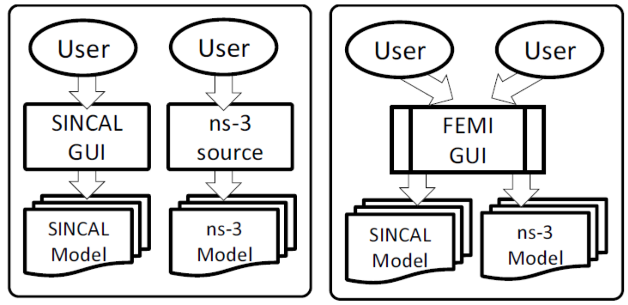

Figure 2. Comparison of the traditional modelling workflow and the FEMI-based workflow.

- Export domain-specific models. Model developers use FEMI GUI to create federation models. The federation model can be converted into multiple domain-specific models for simulation. As aforementioned, this step is necessary because domain-specific simulators normally have their own way of model representation.
- Interact with simulators. At runtime, after the models are generated, it can be loaded into the simulator directly for validation. From FEMI, model developers can control the simulator by starting, pausing and stopping the simulation. The simulation results can be pushed back to FEMI via this interface.

Technically, these services are implemented as RESTful Web Service to cope with the HTML5 Web Application of FEMI. Due to the page limit, the detailed RESTful service specification is omitted.

C. Spatial Support

FEMI has built-in support for spatial objects. Model developers are able to determine the element locations, length, and orientations, etc. via the graphical user interface provided in FEMI. The underlying models will be synchronised automatically if a spatial object is changed.

Dependency recommendation will be performed by analysing the distance between two elements (by taking into account their domains as well). Threshold values can be given by users to automatically trigger the dependency generation. Moreover, areas that model elements can influence (a cabinet is responsible for providing electrical power for a certain area in a city) can also be generated automatically by using methods like the Voronoi diagrams. This provides a convenient way to assess the impacts and consequences of model failures (e.g., how a failure in the power system could potentially influence the operation of a heavily populated area).

IV. PRELIMINARY RESULTS

In order to demonstrate the proposed approach, a training system for crisis management is used. This training system is developed for working with crisis managers by performing What-if Analysis and Consequence Analysis of their decisions [13]. All dynamics of simulated systems are

 ISBN: 978-1-61208-501-2

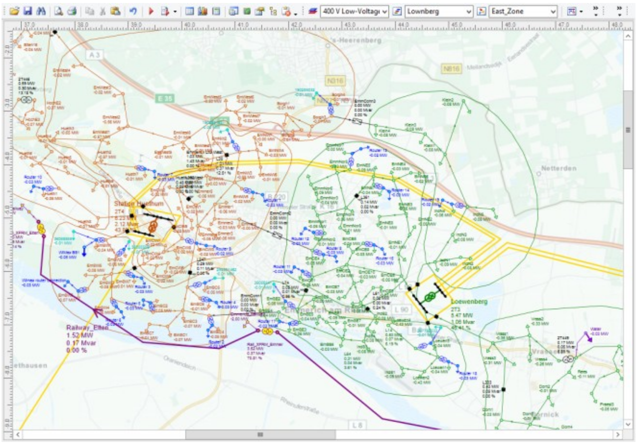

Figure 3. Synthetic electrical distribution network of Emmerich area.

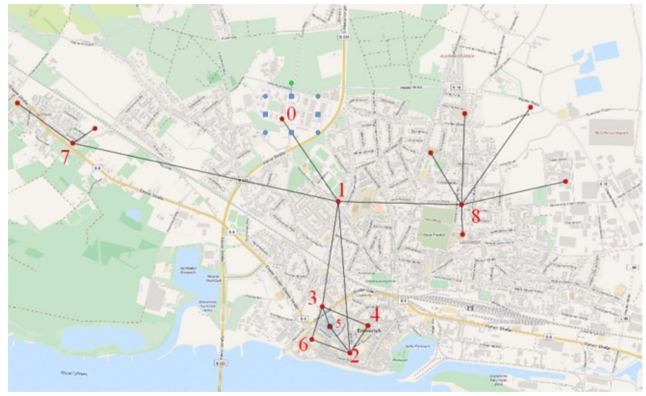

Figure 4. High-level topology of the telecommunication network.

provided by using a spatial-aware federated simulation subsystem.

Two domains are involved in the federated simulation: the power transmission networks and the telecommunication networks. SINCAL is used to perform the power flow calculation and ns-3 is adopted for IP-based Internet traffic analysis. The traditional way to build simulation models is using the tools provided by simulators. For instance, the SINCAL GUI can be used for that purpose and ns-3 models are embedded in the source code of a simulation (see the left part of Fig. 2). Dedicated simulator modules (with limited spatial support) are used to export the destination models. With this approach, the dependencies between models cannot be expressed explicitly. On the right hand side of Fig. 2 is the workflow based on FEMI. Different users, model developers in this case, use the same integrated tool to provide necessary information for building domain-specific models. Dependency modelling is supported out-of-box. Spatial information can be managed by using GIS utilities as explained in Section III.C. With the help of simulator adapters, consistent models can be exported automatically. This is extremely important for federation models that need frequent changes to ensure the model consistency.

The SINCAL model focuses on the distribution network of the given area along with the relevant parameters, as shown in Fig. 3. % (for security reasons we removed the names). The telecommunication routers are included into the SINCAL model as well since they consume electrical power. On the other side, the routers provide the telecommunication services for the given area, which introduce some kind of dependency between the power and the telecommunication systems.

A. Electrical Power System Model

The power system modelled in SINCAL is a synthetic distribution network of the given area. SINCAL is a tool which provides a range of modules for designing, modelling and analysing power models. SINCAL can be used for general load flow analysis, load profile analysis for investigating a daily profile, load development analysis for calculations with load values that vary over time, and contingency analysis for elements malfunction in individual load flow calculations.

The data required to design the model are the physical parameters and the types of the elements. The main elements along with their types are the following:

- Power generation units: conventional generators, renewable energy systems, representation of transmission line equivalent;
- Power connection lines: transmission in high-voltage (HV), distribution in medium-voltage (MV) and low-voltage (LV), railways in HL and MV;
- Electrical loads (end users): consumers of electricity, power supply of telecommunication system, power supply of water system, electrification of transportation system;
- Power transformers: power stations, sub-stations, distribution cabinets/feeders, boost-transformers;
- Static converters: integration of RES to power system, HVDC stations, FACTS, frequency changer.

The model designed in SINCAL is formed by several power elements. The identified physical parameters are specified in Table I. The SINCAL model of power system elements along with the dependent elements are shown in Fig. 3.

B. Telecommunication Network Model

In order to model and simulate telecommunication aspects, we have used the Network Simulation version 3 tool [4]. It is a discrete-event network simulator, provided as free/open source software under a GNU GPLv2 license. Its purpose is to provide an ``open simulation environment for networking research", including IP-based networks and non-IP based communication networks. In the proposed approach, we used ns-3 for modelling the fixed line and mobile telecommunication net-works of our scenarios. However, since we are lacking the information about the real topology of core telecommunication network in the analyzed geographical area, we have made some conceptual and designing assumption while modelling.

It must be noticed, that in our approach we have chosen some examples of simulators that are recognized by the community. For instance, counterparts for NS3 are tools like OMNet++, OPNET, J-SIM, etc. These tools have different advantages and disadvantages (e.g. openness, community support, etc.). However, there were not any particular reasons for choosing NS3, except the fact that we had some experience with that tool before.

First, we assumed that the broadband network to city is provided by node indicated as ``0'' in Fig. 4. The city center is connected via router indicated as "1", that further connects western (router 7), eastern (router 8) and southern parts (router 2) of modelled area.

In southern part of modelled area, there are redundant paths between routers. For instance, traffic from node 5 to node 1 can be established via node 3 or node 2. However, these paths have different characteristics and the link between node 1 and 3 has 20 ms delay in contrast to link between nodes 1 and 2, which has 2 ms delay. In case of the scenario, where node 2 fails, connection will be established via node 3. Therefore, we will observe the degradation of communication quality. As a result, this will disturb the communication between southern areas of modelled area with remaining elements in the city.

The geographical region with close proximity to node 8 has been indicated as urban area, where schools and households are located. The network elements located in this part of the city will generate significant volume of traffic to other services (e.g., banks, hospitals, etc.).

In the ns-3 tools suite, the topology and the configuration of the simulation are provided either in *.py (python) or in *.cc (c/c++) files. Commonly, these files contain following information:

- ns-3 nodes definition (names, types , positions, etc.),
- Communication links definition (data rates and delays),
- Topology definition,
- IP stack installation,
- IP addresses assignment,
- Routing definition,
- Configuration of the application layer.

In ns-3, the term node is used to name an abstract device connected to a network such as end-users hosts, end-systems, routers, switches, hubs etc. Since ns-3 does not focus on Internet technologies only, it is the responsibility of simulation creator to define nodes properly by adding applications, protocols stack, etc. In ns-3 the concept of application is defined as an element that runs the simulation. It is the basic abstraction of a user program, which generates some network traffic.

Currently, ns-3 does not provide GUI that would support modelling process concerning geospatial aspects. There are some projects that aim to provide environments for network topology prototyping. For instance ns-3 topology generator [14] allows the user to use GUI in order to define nodes, communication channels and applications. However, the generated models are represented as C/C++ code, which in some cases may not compile, due to the fact that ns-3 is still under development and the base code changes between releases. Manual changes of the generated code can fix some compilation errors, but user is required to have some expert skills and the changes need to be applied each time topology is generated. Moreover, there is no functionality that will allow users to specify geographical positions of topology elements. Therefore, we believe that FEMI-based workflow is a good way to address these issues and to facilitate model developers with an abstraction layer that will reduce the modelling effort substantially.

C. *Integrated modelling*

For the crisis management training system, we started the modelling work by using the traditional approach as depicted in Fig. 2. Soon we noticed that it does not scale and is very time-consuming to maintain the model consistency with different tools.

With the proposed approach, for dependent elements like routers that exist in both domain-specific models, only one instance needs to be managed. By specifying the corresponding domains, separate instances will be generated automatically in the target model. This feature substantially reduces the efforts to improve the model consistency. For instance, it happens quite often in preparing federation models for training that the name of a router in the ns-3 model is changed while the reference in the SINCAL model still has the old name. This causes unexpected runtime behaviours of the federated simulation system and it can be avoided with the proposed approach. Unfortunately, since FEMI is still under development and the GUI part is still not complete, we will show screenshots in the forthcoming publications. In addition, formal ontology is also considered as a common vocabulary between different domains to facilitate a consistent modelling. However, at this stage of our development cycle we still rely on consistent naming convention when connecting the same elements in different simulators. Also some specific mapping between FEMI core system and the simulators (e.g. bandwidth or delay of telecommunication links) are still hardcoded into adapters logic.

V. CONCLUSION

This paper briefly presented the motivation, ideas and benefits of using an integrated modelling environment for spatial-aware federated simulations. The motivation of this work laid mainly in the extremely high overhead in maintaining model consistency in traditional federated simulation modelling workflow. Moreover, limited GIS support makes it difficult, if not impossible, to facilitate modelling and visualising spatial objects and dependency. Based on this consideration, the FEMI system was illustrated. The software architecture, the interface specification and the spatial support were elaborated to provide a high-level overview of this system. It is still a work in progress and currently under development. One of the core parts of FEMI is the simulator adapter. Implementing such an adapter is a challenging task, because the developers need to know the technical details of communication interface provided by FEMI and simulators. This is one of the drawbacks of the proposed approach. Due to the heterogeneity of different domain specific simulators, this can be an effort-intensive task.

In the future, we plan to accelerate the development work of FEMI and make it publicly available as a cloud platform for modelling federated simulations. Collaborative modelling is also one of the features that we want to address in FEMI so that different model developers can work together within one Web-based platform. Formal ontologies will be adopted as common vocabularies to facilitate a consistent modelling process. Finally, the envisioned European Scenario Database [15] can be connected with FEMI to provide an end-to-end solution for generating and reusing federation models.

ACKNOWLEDGMENT

This research leading to these results was funded by the European Commission within the Seventh Framework Programme project CIPRNet (grant agreement N° 312450) and PREDICT (grant agreement N° 607697). The authors thank all of the project partners for many constructive discussions and useful suggestions.

REFERENCES

[1] E. Rome, P. Langeslag, and A. Usov, “Federated modelling and simulation for critical infrastructure protection,” in Networks of Networks: The Last Frontier of Complexity. Springer, 2014, pp. 225–253.

[2] A. Usov, C. Beyel, E. Rome, U. Beyer, E. Castorini, P. Palazzari, and A. Tofani, “The DIESIS Approach to Semantically Interoperable Federated Critical Infrastructure Simulation,” in Advances in System Simulation (SIMUL), 2010. IEEE, 2010, pp. 121–128.

[3] SimTec, “PSS SINCAL,” 2015. [Online]. Available: http://www.simtec.cc/sites/sincal.asp

[4] NS-3 Consortium, “NS-3 Network Simulator,” 2015. [Online]. Available: http://www.nsnam.org

[5] IEEE, “IEEE 1516-2010 - Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules,” IEEE, Tech. Rep., 2010.

[6] J. Mart´ı, C. Ventura, J. Hollman, K. Srivastava, and H. Juarez, “I2sim modelling and simulation framework for scenario development, training, and real-time decision support of multiple interdependent critical infrastructures during large emergencies,” in NATO (OTAN) MSG-060 Symposium on How is Modelling and Simulation Meeting the Defence Challenges out to, vol. 2015, 2008.

[7] S. Rinaldi, J. Peerenboom, and T. Kelly, “Identifying, understanding, and analyzing critical infrastructure interdependencies,” Control Systems, IEEE, vol. 21, no. 6, 2001, pp. 11–25.

[8] A. Nieuwenhuijs, E. Luiijf, and M. Klaver, “Modeling dependencies in critical infrastructures,” in Critical Infrastructure Protection II. Springer, 2008, pp. 205–213.

[9] A. D. Giorgio and F. Liberati, “A bayesian network-based approach to the critical infrastructure interdependencies analysis,” Systems Journal, IEEE, vol. 6, no. 3, 2012, pp. 510–519.

[10] IEC, “IEC 61970-301 Energy management system application program interface – Common Information Model base,” International Electrotechnical Commission, Geneva, Switzerland, ISO 27019-2013, 2013.

[11] S. Tilkov and S. Vinoski, “Node.js: Using javascript to build high-highperformance network programs,” IEEE Internet Computing, vol. 14, no. 6, 2010, pp. 80.

[12] P. Crickard III, Leaflet.js Essentials. Packt Publishing Ltd, 2014.

[13] CIPRNet, “The EU CIPRNet Research Project,” 2013. [Online]. Available: http://www.ciprnet.eu

[14] University of Strasbourg, “Topology Generator Project,” 2016. [Online]. Available: https://www.nsnam.org/wiki/Topology%5FGenerator.

[15] J. Xie, M. Theocharidou, and Y. Barbarin, “Knowledge-driven scenario development for critical infrastructure protection,” in Critical Information Infrastructures Security - 10th International Workshop, CRITIS 2015, Berlin, Germany, October 5-7, 2015, ser. Lecture Notes in Computer Science. Springer, 2015, pp. 91-102.