Graphical formalisms for modelling critical infrastructures

A. Bondavalli, P. Lollini & L. Montecchi
Department of Systems and Computer Science, Florence University, Florence, Italy

Abstract

Modelling and simulation are well-suited approaches to analyse critical infrastructures (CIs), providing useful insights into how components' failures might propagate along interconnected infrastructures, possibly leading to cascading or escalating failures, and to quantitatively assess the impact of these failures on the service delivered to users. This chapter focuses on the usage of graphical formalisms for modelling and simulation of CIs. It first identifies and motivates the main requirements that a modelling and simulation framework for CI analysis should have. Then, it provides an overview of the available graphical formalisms, discussing how they have been used in the literature for CI analysis and assessing the extent to which they actually meet the identified modelling and simulation requirements. The second part of the chapter investigates how a subset of the identified requirements are actually met adopting a specific graphical modelling formalism, the Stochastic Activity Networks formalism, which has been extensively used by the authors of this chapter in past European FP6 projects dealing with CI analysis.

Keywords: Critical Infrastructures, Modelling and Simulation, Graphical Formalisms, Modelling Requirements, Stochastic Activity Networks

1 Introduction

Critical infrastructures (CIs) are complex and highly interdependent systems, networks and assets that provide essential services in our daily life. They span a number of key sectors, including energy, finance, authorities, hazardous materials, telecommunications, information technology, supply services and many others. In view of this widely recognized criticality, it is paramount that they be reliable and resilient to continue providing their essential services. Hence, there is the need (i) to build such CIs following sound engineering design principles, (ii) to protect them against both accidental and malicious faults and (iii) to evaluate them to assess their degree of resilience/trustworthiness.

WIT Transactions on State of the Art in Science and Engineering, Vol 54, © 2012 WIT Press
www.witpress.com, ISSN 1755-8336 (on-line)
doi:10.2495/978-1-84564-562-5/04

Modelling and simulation play a key role in CI protection, since experimenting on such critical systems is often costly or dangerous. Because of the complexity and interconnectedness of such systems, modelling and simulating CIs is a well-recognized challenge, especially if the interactions between different infrastructures are to be considered. Several approaches to master this complexity have been proposed in the past literature. In this chapter, we focus on modelling and simulation approaches that are supported by *graphical formalisms*. Besides surveying the available graphical formalisms, we will also inspect their capability to satisfy a set of basic requirements that a modelling and simulation framework for CI analysis should satisfy, both considering the works available in the literature and basing on the experiences gained by the authors of this chapter in recently ended European projects.

The rest of the chapter is organized as follows. Modelling and simulation requirements are identified and discussed in Section 2. Section 3 surveys the available graphical formalisms and discusses, from the authors' perspective, the extent to which they actually meet the basic requirements. In Section 4, we deeply investigate how a subset of the basic requirements provided in Section 2 are actually met by a specific modelling formalisms, the Stochastic Activity Networks (SAN) formalism, which has been extensively used by the authors of this chapter in past European FP6 projects. Finally, conclusions are drawn in Section 5.

2 Requirements for CI modelling and simulation

Requirements for CI modelling and simulation are strictly related to the objectives of the analysis. Among the approaches proposed in the past literature, some of them focus on the interdependencies among infrastructures, and they elaborate integrated approaches capable of capturing the specific characteristics of the different CIs as well as their relationships. Complexity and heterogeneity can be overcome by modularity and composition, using multi-formalism approaches (e.g., see [1]). For what concerns model solutions, the concept of *federated simulation* (e.g., see [1]) has emerged as a viable solution to the simulation of large and interconnected systems. Such approach aims at creating a composable simulator, supporting interoperability among separately developed simulators through a unified programming interface. Following this 'system of systems' philosophy, IEEE has defined the High-Level Architecture (HLA) specification [3] with the aim to provide a standardized interface. Works exist in literature that define requirements for the construction of a 'universal' CI modelling environment following such approach (e.g., see [2] and [5]). The main requirements for integrated modelling and simulation of CIs are as follows:

R1: *The integrated simulation environment should be able to represent physical, cyber, geographical and logical interdependencies.*

Four kinds of interdependencies between CIs have been defined in literature [6]. Such a universal simulation environment should take into account all these interactions and their effects.

R2: *The integrated simulation environment should be able to represent and simulate cascading effects.*
A cascading failure occurs when a disruption in one infrastructure causes the failure of a component in a second infrastructure, which consequently causes a disruption in the second infrastructure. Such dynamics should be taken into account in an integrated simulation approach.

R3: *Modelling and simulation solutions for CI analysis should provide a method for accommodating different simulation methodologies.*
Different infrastructure models may leverage different simulation methodologies. This requirement highlights the necessity for an approach to mediate the differences among simulation methodologies, for example, between continuous and discrete simulation methodologies.

Besides defining the requirements for an integrated modelling and simulation approach dealing with 'system of systems', it is paramount to analyse the requirements for modelling single, isolated infrastructures in order to faithfully represent the specificity of each domain. To the best of our knowledge, two of the most detailed works in identifying and discussing these modelling requirements are [5], which aims at evaluating the elements to be included in a composable simulator, and [7], which is specifically tailored to the electric power domain. Based on these works and on the experiences gained by the authors in two past European projects addressing CIs (see Section 4), we identified the following basic modelling requirements:

R4: *The formalism should support the modelling of large and hierarchically structured CIs in a convenient way.*
Many systems in general, and CIs in particular, have a natural hierarchical structure with a large number of components belonging to different levels and arranged in a treelike structure. At a certain level of detail such systems are typically composed by many similar components having the same logical structure, which can be grouped on the basis of their similarities. From a modeller's perspective a key need is to have some modelling features that facilitate the model construction exploiting such similarities. This would also provide benefits in model maintainability, readability and reusability.

R5: *The formalism should support the representation of discrete, continuous and hybrid state, using a compact representation.*
Most CIs are hybrid state systems, in the sense that part of their state-space is continuous and another part is discrete. The continuous state-space is usually related to the physical aspects of the system, which are governed by complex mathematical relations; the discrete part is instead related to the control layer, which comprises a set of operational states and decision policies.

R6: *The formalism should support the interaction with external tools and functions, which may properly capture the details of specific parts of the system.*
This requirement is also related to the application of the federated simulation approach described above. In general, being able to interact with external tools and functions allows the model to have access to external data, for example, data collected by experiments on the real infrastructure.

R7: *The formalism should support the definition and evaluation of both dependability and performance-oriented metrics.*
CIs are often subject to market constraints and must therefore achieve some predefined levels of performance in addition to fulfil their dependability requirements. It is the case, for example, of the electricity market within the electric power system (EPS) or QoS levels in networking systems. Therefore, an overall evaluation of a (critical) infrastructure is likely to be based on both dependability and performance-oriented metrics. The formalism should allow the specification of many different measures of performance, dependability and performability in a unified manner.

In the following section, we give an overview of the graphical formalisms introduced in the literature, and we shortly discuss how they fulfil the identified set requirements both concerning the 'system of systems' approach (*requirements* R1–R3) and an individual infrastructure analysis (*requirements* R4–R7).

3 Graphical formalisms for CI modelling and simulation

Several approaches to CI modelling and simulation have been adopted in the literature, each having different levels of detail, modelling power, user-friendliness and computational efficiency. The works in [8] and [9] provide a general understanding of common methods for CI analysis, including visualization and data-presentation techniques, while a specific survey focused on modelling and simulation can be found in [10]. For what concerns existing tools for CI analysis, a large collection of them is reviewed in [11] and [12].

Depending on the formalism, graphical information plays a different role in model construction and evaluation. Essentially, the use of graphical information to aid CI analysis may be grouped in few main areas, which are detailed in the following.

3.1 Graph-based techniques

Many approaches to CI modelling are based on graph-analysis techniques. In such approaches, the physical topology and configuration of the infrastructure are mapped to some kind of graph, which can then be analysed to reveal useful information about the system. Through this representation, many of the already available graph-analysis techniques can be used to analyse the behaviour of the modelled infrastructure(s). For example, using this representation, resource

allocation problems may be formulated in terms of graph colouring problems, while some reliability properties may be analysed through clique problems [9].

To perform assessments with respect to faults or external attacks, CIs are often modelled as networks, and then nodes are progressively removed to evaluate the possible cascading effects on the system. These kinds of analyses are used to compare infrastructure designs and topologies, for example, showing the maximum number of random attacks that a certain topology may handle before becoming disconnected (e.g., see [13]).

Although these analyses may provide useful insights on the infrastructure properties, it is often necessary to take into account also other aspects of the system. Network flow approaches are used to model resource requirements and utilization among different infrastructures. In such paradigm, interdependent infrastructures are viewed as networks, with movement of commodities (i.e., material, electric power etc.) corresponding to flows and with services corresponding to a desired level of these flows. Approaches based on network flow are easily modelled using supply–demand graph; in such kind of graphs, nodes are seen either as supply, transhipment or demand nodes, while arcs represent links through which commodities flow from producers (supply) to consumers (demand) nodes. Nodes may be both producers and consumers at the same time; for example, a gas alimented power generator supplies electric power, but it demands natural gas to perform its function. Supply–demand graphs have been used, for example, in [14] to identify the telecommunication components which are more vulnerable to failures of power components within a certain CI design.

Different mathematical formalisms may be associated to supply–demand graphs, leading to many variants of such approach. In [15], link capacities are taken into account, considering both deterministic and stochastic values; in [16], nodes may have buffers to hold storable resources, for example, water or gas.

3.2 Petri Nets (PNs)

PNs and their extensions are graphical modelling formalisms that are widely used in dependability analysis. Although they have a simple graphical representation, they provide a great modelling power and are therefore well suited for the modelling of complex systems like CIs. Many variants of PN formalisms exist, which may have different properties and modelling power.

In [17], a set of CIs are modelled at a very high level of abstraction, focusing on interdependencies between them; then it is shown how invariant analysis on the PN model can be used to identify vulnerable elements in the scenario. The authors of [18], using the Generalized Stochastic Petri Net (GSPN) formalism, define some useful primitives to model common mode faults and cascading effects in CIs, using an actual power blackout as motivating example. In [19], a GSPN model is developed to evaluate the impact of a potential intrusion due to a cyber attack on the Supervisory Control and Data Acquisition (SCADA) system, which is in charge of controlling and monitoring the EPS. There are two submodels in the PN model, a firewall model and a password model, which are

instantiated based on the configuration of the internal SCADA network and its possible access points. A combined modelling approach in the evaluation of the interdependencies between the electric power infrastructure and its SCADA system has been developed in [20], where the quantification is achieved through the integration of two models. The first is a SAN model, which concentrates on the structure of the power grid and its physical quantities; the second is a Stochastic Well-Formed Net (SWN) model, which concentrates on the algorithms of the control system and on the behaviour of the attacker. The scenario modelled in such work considers a situation in which a load shedding activity is needed to re-establish the nominal working conditions upon an electrical failure, but the control system is not working properly due to a Denial of Service (DoS) attack. Finally, in [21], PNs have been employed in the evaluation of pricing issues related to congestion in deregulated power market systems.

3.3 General simulation environments

A large collection of simulation environments exists for CI analysis, which can essentially be categorized in single domain and multiple domain simulators. The electric power infrastructure, together with telecommunication networks and transportations, has been the focus of development of domain-specific simulators, featuring many simulation tools having different granularity [12].

Graphical facilities play a key role in simulation environment: first, they allow the user to focus on the high-level details of the model and simulation experiments; next, a graphical simulation tool provides by its very nature a graphical representation of the model, which may be of invaluable benefit for model maintainability. Finally, user-friendliness may make the success of simulation tools: a well-designed graphical environment can provide cost-effective, integrated and automated support of simulation model development throughout the entire modelling and simulation life cycle.

Visualization refers to the discipline that ‘focuses on helping people explore or explain data, typically through software systems that provide static or interactive visual representations’ [8]. Visualization techniques may focus on graphical representation of the model itself, for example, 3D representation of infrastructure entities [22], or map overlay of multiple layers, which may include other infrastructure models or even Geographic Information System (GIS) data [23]. Other visualization techniques focus on the presentation of simulation results, contributing to the identification of correlation between the parameters of a system or the detection of logical interdependencies. Just to cite a few, these techniques include function fitting, overlaying, shading, spectral planes, interactive (and continuous) rotation of 3D displays [9].

3.4 Agent-based modelling and simulation

The agent-based paradigm is a promising approach to software development, which has been proven particularly useful in modelling and simulation of CIs.

It consists of a bottom-up approach to manage system complexity, in which the simulator is built as a population of interacting, intelligent *agents*. An agent is 'an autonomous system (software and/or hardware) that is situated in an environment (possibly containing other agents) and acts on it in order to pursue its own goals, and is often able to learn from previous experiences' [10]. Each agent is an individual entity with location, capabilities and memory. Interaction between them produces an *emergent behaviour,* that is, a behaviour which is not predictable by the knowledge of any single agent [24]. Using such approach, a simulator is developed, where an agent may model physical components of infrastructures, decision policies or, possibly, the external environment [4,24].

As other modelling techniques, the agent-based paradigm can be applied at different levels of detail, which are sometimes referred to as micro- and macro-agent-based simulation [25]. The micro-agent-based approach uses a bottom-up approach modelling for every single component of an infrastructure, putting them successively together to simulate the whole infrastructure(s). The macro-agent-based simulation represents a whole infrastructure with a single agent, hiding the implementation details from the other agents. Using such approach, it is also possible to apply the federated simulation approach, leaving the physical, detailed simulation of each infrastructure to some specific sector tool controlled by an associated agent and expose only a predefined interface to other agents.

In addition to visualization techniques that are not specific to agent-based modelling, but may be employed in any simulation tool, this approach is often supported by graphical facilities to aid the definition and development of agent-based simulators. The authors of [26] define a graphical way to represent entities and interdependencies in complex systems composed of different infrastructures, using Unified Modelling Language (UML) as graphical formalism; such entities are then mapped to one or more agents in the simulation environment. The example scenario takes into account a Civic Emergency Management system and its dependence on power grid (for the information system functionality), on communication network (for communications) and on transportation network (for emergency operations). Moreover, some specialized agent-based frameworks have built-in graphical capabilities to define the interconnection between the agents or even their behaviour. For example, the Repast Toolkit [27] allows the graphical specification of agents' behaviour, using graphical primitives like *task, decision, join, loop.*

3.5 Discussion of requirements

In this section, we discuss how *requirements* R1–R7 are actually fulfilled by the available formalisms and analysis approaches that have been proposed in the literature. Such evaluation is based on the authors' perception of the average capabilities of the formalisms belonging to each category and on their usefulness with respect to the identified requirements. The overall results are summarized in Table 1, where we denote with '+' the requirements that can be more easily achieved within the different categories.

Table 1: Evaluation of formalisms with respect to requirements

	R1	R2	R3	R4	R5	R6	R7
Graphs	+	+		+			
Petri nets	+	+		+			+
Simulation	+	+			+	+	
Agents	+	+	+			+	

Graph-based approaches are generally good at defining the hierarchical structure of the system (R4) and the interdependencies that exist between infrastructures (R1), as graphs are a natural way to represent relations between elements. For the same reason, cascading failures may be represented as well (R2). The limitations of graph-based approaches consist in their reduced scalability and their limited modelling power. Graphs may be used by other advanced formalisms to represent the structure of the system or the analysed scenario, but with this exception they are practically unable of integration with other modelling tools (R3 and R6). Graph-based formalisms are often tailored to a specific measure or analysis type and do not allow the definition of different measures (R7).

Formalisms belonging to PN category are usually also well suited to the definition of performance and performability measures (R7), but they have similar limitations of generic graph-based approaches. PN models allow the modelling of interdependencies and cascading effects at a very high level of detail, but representing the hierarchical structure of the whole system, taking into account of all the interdependencies, may be a difficult task and the scalability of the model is often a limiting factor. The extent to which they are able to represent both discrete and continuous states (R5) highly depends on each individual PN formalism, but with some exceptions, they are usually tailored to model discrete state systems. Integration with other tools and differing simulation methodologies (R3 and R6) is not possible, with the exception of few PN formalisms.

Simulation packages may easily represent non-discrete system states (R5), and to some extent they allow the modelling of interdependencies and cascading effects (R1 and R2). Integration with external tools (R6) is possible, although it often requires a significant effort to be achieved. The HLA and other similar initiatives are supposed to facilitate this task in the long run. Simulation environments may allow the evaluation of very complex measures, but they are usually able to evaluate a limited predefined set of them.

Agents perform particularly well in satisfying the requirements related to the integrated modelling and simulation approach (R1–R3): by their nature, they model the system as a population of interdependent autonomous subsystems (i.e., agents). External tools can be usually integrated quite easily in agent-based frameworks (R6), thanks to the macro-agent and federation approach. Agent-based simulation frameworks have the same limitations that arise in other simulation frameworks for what concerns the available measures that can be evaluated.

Although there is no formalism category that, as a whole, is capable to fulfil all the identified requirements, some specific formalisms belonging to specific categories feature more advanced capabilities that can be used to profitably model complex systems as CIs. An example belonging to the PN class is the SAN formalism, which provides the modeller with some primitives that can be profitably exploited to fulfil the identified requirements, thus overcoming the limitations of most of the other PN-based models. In the next section, we provide more details on such capabilities, also showing in particular how to exploit some particular SAN features to model the hierarchical system structure in a convenient way (R4) and to facilitate the integration of external tools and functions (R6).

4 Practical experiences in modelling CIs: meeting the requirements with SAN

SAN [28] formalism is a powerful and flexible extension of PNs, and for its characteristics it has been extensively applied to model and analyse complex CIs. As discussed in the following, the formalism meets the whole set of identified requirements. The SAN capabilities in representing the different kind of interdependencies (*requirement* R1) have been discussed in [7], focusing on the cyber and physical interdependencies in the electric power domain. The same work has also inspected their use for capturing cascading failures (*requirement* R2) from the information control system towards the controlled electric power grid, which can finally lead to blackout phenomena. The accommodation of different simulation methodologies (*requirement* R3) could be supported by specific SAN primitives (input and output gates), general functions written in languages like C that could trigger different simulation methodologies. The representation of both discrete and continuous states (*requirement* R5) is another SAN feature: the SAN formalism supports continuous valued tokens, thanks to a special primitive called 'extended place' that allows token of more complex data types to be included in the model. Each extended place is assigned a 'type' (much like in ordinary programming languages), and it is allowed to hold tokens of such type. The definition and evaluation of both dependability and performance-oriented metrics (*requirement* R7) is fully met resorting to the Performance Variable (PV) reward model, which can be used to represent either dependability or performability measures.

In the following we will further discuss the two remaining *requirements* R4 and R6, instantiating them in the CRUTIAL and HIDENETS contexts, respectively. For each requirement, we analyse the useful features of the SAN formalism, and we show how they have been exploited in the projects to fulfil each requirement. The research activities on the usage of SAN for modelling CIs, started within these two projects, are now carried on within the ongoing Italian project PRIN [29] DOTS-LCCI, which focused on the analysis and evaluation of Large-Scale Complex Critical Infrastructures (LCCI).

4.1 CRUTIAL and HIDENETS: a brief introduction

The European project CRUTIAL [30] addressed new networked systems based on information and communication technology for the management of the electric power grid. A major research line of the project focused on the development of a model-based methodology for the dependability and security analysis of the power grid information infrastructures. One of the approaches pursued in CRUTIAL was a model-based quantitative support for the analysis and evaluation of critical scenarios in EPS, as incrementally documented in [7,31,32].

The European project HIDENETS [33] addressed the provisioning of available and resilient distributed applications and mobile services in highly dynamic environments characterized by unreliable communications and components. A set of representative use-case scenarios were identified, each one composed by different applications (mostly selected from the field of car-to-car and car-to-infrastructure communications), different network domains, different actors and characterized by different failure modes and challenges. As incrementally documented in [34] and [35], the authors of this chapter focused on the QoS analysis of a dynamic, ubiquitous Universal Mobile Telecommunication System (UMTS) network scenario, which comprised different types of mobile users, applications, traffic conditions and outage events reducing the available network resources.

4.2 On the usage of SAN to match requirement R4

Let us consider the EPS analysed within CRUTIAL, which is composed of two cooperating infrastructures: the Electric Infrastructure (EI) for electricity generation and transportation, and its Information Technology Based Control System (ITCS) in charge of monitoring and controlling the EI physical parameters and of triggering appropriate reconfigurations in emergency situations. A complete view of the EPS logical structure at regional level can be found in [7] and is illustrated in Figure 1.

In the lower part of Figure 1, we have depicted the main logical components that constitute the EI: generators (N_G), loads (N_L), substations (N_S) and power lines (A_L). From a topological point of view, the power transmission grid can be considered like a network, or a graph, in which the nodes of the graph are the generators, substations and loads, while the arcs are the power lines. In the upper part of Figure 1, we have depicted the logical structure of a regional ITCS, that is, the part of the information control system controlling and operating on a region of the transmission grid. The components LCS (Local Control System) and RTS (Regional Tele-Control System) differ for their criticality and for the locality of their decisions, and they can exchange grid status information and control data over a (public or private) network (ComNet component). LCS guarantees the correct operation of a node (generator, substation or load) and reconfigures the node in case of breakdown of some apparatus. RTS monitors its assigned region in order to diagnose faults in the power lines. In case of breakdowns, it chooses the most suitable corrective actions to restore the functionality

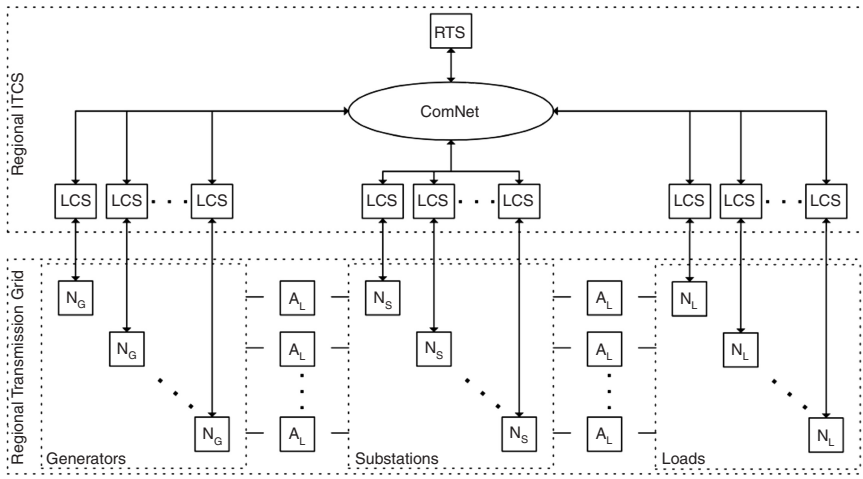

Figure 1: Logical structure of a regional transmission grid, with the associated information control system.

of the grid. When considering a large portion of the grid, we have to deal with a huge number of components that need to be modelled, replicated and composed to form the hierarchical structure of the whole EPS, as shown in Figure 1. A way to proceed could be to manually duplicate the template models representing the different basic components of EI (generators, loads, substations, power lines) and ITCS (LCS, RTS), to manually assign them a specific parameters setting and finally to compose them obtaining the model for the overall system. This modelling process can be very expensive in terms of time and very error prone, so we would like to have a modelling formalism that *facilitates the construction of the overall model allowing model composition and automatic model replication*. The hierarchical structure should be defined by automatically replicating the basic template models and composing them as needed. For example, the model that represents a generic power line needs to be replicated to obtain all the necessary A_L components of the grid. In the same way, the basic LCS model associated to a node of the grid needs to be replicated to obtain all the necessary LCS components. Finally, the model for the overall system should be obtained through composition of the different replicated submodels.

The *Replicate/Join* composed model formalism (see [36] and [37]) for SAN actually provides very useful supports for building hierarchical models, allowing the modeller to define a composed model as a tree in which the leaves are the submodels and each non-leaf node is a Join or a Replicate node. The root of the tree represents the complete composed model. A Join is a general state-sharing composition node used to compose two or more submodels, and it may have other Joins, Replicates or other submodels defined as its children. A Replicate is a special case of the Join node used to construct a model consisting of

a number of *identical copies* of a submodel. Since all the copies are identical, the resulting model has the same behaviour of the model where all the copies of the same submodel are composed using a Join node. A Replicate node has one child, which may be another Replicate, a Join or a single atomic or composed model. The modeller may also specify a set of state variables to be held in common among all replicated instances of the submodel.

Although Replicate can be profitably used to automatically build replicas of the same model, it has the limitation that all the replicas generated in this way are *anonymous*, as they are all identical copies of the same submodel. Conversely, the replicas within the CRUTIAL model needed to be *non-anonymous* (i.e., distinguishable), as each of them had a specific role and position within the electric grid as well as a different setting of parameters. However, exploiting the Replicate compositional operator and the ability to define shared places, it is possible to create *non-anonymous* replicas as well. In detail, we defined a template SAN model that, once plugged (i.e., added) into a generic model that needs to be replicated, allows to distinguish between the different replicas assigning each replica a different index, represented by the number of token that the replica holds in a certain place.

The SAN model implementing this specific feature is shown in Figure 2. Let us consider the A_L components of Figure 1 (the power lines); if m is the total number of power lines in the system, the model corresponding to the A_L component needs to be anonymously replicated m times, using the Replicate compositional operator. The number of tokens in the local place ALindex represents the index of the replica. This place is set by the output gate setIndex when the immediate activity setupIndex completes, which is defined as follows: ALindex → Mark() = (m-ALcount → Mark()) − 1. The place Start is initialized with one token.

The common place ALcount is shared with all the replicated instances, and it is initialized with m tokens. The immediate activities setupIndex of the replicated instances are all enabled in the same marking at time 0. Thus, the first instantaneous activity setupIndex that completes removes one token from places ALcount and Start, and then the code of setupIndex is executed, thus setting to 0 the place ALindex of the same instance. In the same way, the second activity setupIndex that completes will finally set 1 token in the associated place ALindex and so on. Therefore, at the end of this (instantaneous) 'initialization' process, a different index (ALindex → Mark()) will be associated to each instance of the model, thus obtaining non-anonymous replicas of the A_L component.

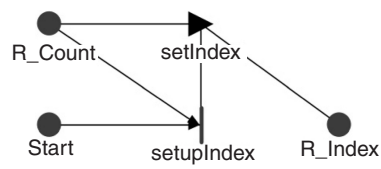

Figure 2: SAN plugin for the indexing of replicas.

4.3 On the usage of SAN to match requirement R6

As stated in Section 2, federated simulation is envisioned as the most promising approach, as CIs are highly dependent on each other, and a vast collection of domain-specific tools are available for CI modelling and simulation. A similar approach has been used also in the HIDENETS context, where the evaluation is performed using a composed simulator, namely, a simulated SAN model in which the mobility aspects are federated to an external vehicular mobility simulator. In fact, such dependency exists between transportation infrastructure and the analysed UMTS networking system, since terminals mobility may heavily affect the QoS metrics. Therefore, we felt within the project that a detailed modelling of the mobility aspects was paramount and would deserve the *integration of an ad hoc mobility simulator into the modelling process itself.* The output of this simulator was then exploited to refine the estimation of the cell load factor increment produced by each service request, thus obtaining a more detailed and faithful model of the UMTS network. Basically, a particular SAN atomic model, called TraceParser, was added to the UMTS network model, having the tasks of executing the external mobility simulator tool, progressively read the trace produced by it and keeping the SAN simulation in sync with the time steps specified in the trace file.

The SAN formalism allows the modeller to include C++ code inside input and output gates. Moreover, while building SAN models, we can define custom functions for the model using C++ header files and libraries. User-defined functions can be extremely useful when trying to make modular models, or if multiple elements within the model, such as SAN output gates, are performing similar operations. The ability to execute C++ code can also be used to call external applications, in this case to execute the mobility simulator, which will generate as output a trace in textual format. The basic TraceParser atomic model is shown in Figure 3.

Although the model developed in HIDENETS is more complex as it contains some features specific for that use case, we provide here a general parser model, which can be used to read a generic trace from the SAN model. In its simplest version, the TraceParser atomic model consists of four places and two activities. Nodes is an extended place which can hold an array of coordinates (i.e., an array

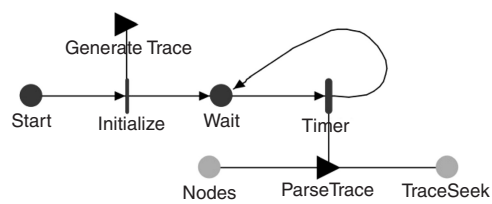

Figure 3: The TraceParser atomic model, which performs the parsing of an external trace.

of structured variables having two float fields, *x* and *y*), and it is used as interface with the replicated models representing the UMTS users; TraceSeek is an extended place which is used to remember the last position that was read in the trace file. Initially one token is held in place Start, thus enabling activity Initialize. Trace generation takes place in the output gate GenerateTrace, thanks to the following call which runs the mobility simulator: system("java -jar VanetMobiSim.jar scenario.xml"). The firing time of the activity Timer is deterministic, and it is set to the length of the time step used in the input trace. In this way, the trace is read incrementally, keeping the simulation time synchronized with the sampling time specified in the trace. The actual parsing of the trace is performed in the output gate ParseTrace, whose function is the following:

```
FILE *ptrFile; int iNode = 0; float fTime = 0, x = 0,
y = 0;
ptrFile = fopen("mobility.trace","r");
for(int i = 0; i , UserCount; i++) {
      fseek(ptrFile, TraceSeek→Mark(), SEEK_SET);
      fscanf(ptrFile," #%d %f %f %f",&iNode, &fTime,
        &x, &y);
      Nodes→Index(iNode) →x→Mark() 5 x;
      Nodes→Index(iNode) →y→Mark() 5 y;
      TraceSeek→Mark() = ftell(ptrFile); }
fclose(ptrFile);
```

The function opens the trace file in the traditional way; then for each user in the model (as specified by the global variable UserCount) the new position is parsed from the trace, using the fscanf function. Together with the new position, the node index is also parsed from the trace, and it is then used to map the new coordinates to a specific replica of the model representing each user. In this way, thanks to non-anonymous replicas, parameterization and the use of structured data types, the new coordinates are easily forwarded to each atomic model instance. The position (in bytes) in the trace file is then saved into the place TraceSeek, in order to resume the parsing on the next iteration.

5 Conclusions

This chapter has addressed the usage of graphical formalisms for the modelling and simulation of CIs. A list of basic modelling and simulation requirements for CI analysis has been provided and discussed. Then, the available graphical formalisms have been surveyed and inspected to understand the extent to which they actually meet the identified requirements. It has been shown that each graphical formalism category is particularly suited to fulfil a subset of the identified requirements. Finally, it has been shown how the SAN features can be

profitably used to meet the modelling requirements, concretely discussing some of them in the context of past FP6 European projects.

Acknowledgements

The authors acknowledge the support given by the European Commission to the research projects CRUTIAL and HIDENETS. This work has been partially supported by the Italian Ministry for Education, University, and Research (MIUR) in the framework of the Project of National Research Interest (PRIN) 'DOTS-LCCI: Dependable off-the-shelf based middleware systems for Large-Scale Complex Critical Infrastructures' [29].

References

[1] Flammini, F., Vittorini, V., Mazzocca N. & Pragliola, C., A study on multiformalism modeling of critical infrastructures, Lecture Notes in Computer Science, vol. 5508, pp. 336–343, 2009.

[2] Casalicchio, E., Galli, E. & Tucci, S., Federated agent-based modelling and simulation approach to study interdependencies in IT critical infrastructures. *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, Chania, Greece, October 22–24, 2007.

[3] IEEE *Standard for Modelling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*. IEEE Std. 1516, Institute of Electrical and Electronics Engineers, New York, 2000.

[4] Tolone, W.J., *et al.*, Enabling system of systems analysis of critical infrastructure behaviors. *Proceedings of the Third International Workshop on Critical Infrastructure Security (CRITIS08)*, Frascati, Italy, October 24–35, 2008.

[5] Flentge, F., *et al., Catalogue of Requirements for SYNTEX*, Integrated Risk Reduction of Information-based Infrastructure Systems (IRRIIS) project, Deliverable available at the following url: http://www.irriis.org/File0475.pdf?lang=2&oiid=8996&pid=572.

[6] Rinaldi, S.M., Peerenboom, J.P. & Kelly, T.K., Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine*, vol. 21, no. 6, pp. 11–25, 2001.

[7] Chiaradonna, S., Lollini, P. & Di Giandomenico, F., On a modelling framework for the analysis of interdependencies in electric power systems. *Proceedings of the IEEE/IFIP 37th International Conference on Dependable Systems and Networks (DSN 2007)*, Edinburgh, UK, June 25–28, 2007.

[8] Bloomfield, R., Chozos, N. & Nobles, P., *Infrastructure Interdependency Analysis: Introductory Research Review*. Produced for CPNI, TSB and EPSRC, under contract NSIP/001/0001, 2009, http://www.csr.city.ac.uk/projects/cetifs/d422v10_review.pdf.

[9] Ghorbani, A.A. & Bagheri, E., The state of the art in critical infrastructure protection: A framework for convergence. *International Journal of Critical Infrastructures*, vol. 4, pp. 251–244, 2008.

[10] Rigole, T. & Deconinck, G., A survey on modelling and simulation of interdependent critical infrastructures. *3rd IEEE Benelux Young Researchers Symposium in Electrical Power Engineering,* Ghent, Belgium, April 27–28, 2006.

[11] Pederson, P., Dudenhoeffer, D., Hartley, S. & Permann, M., *Critical Infrastructure Interdependency Modelling: A Survey of U.S. and International Research*, Idaho National Laboratory (INL), Technical Report, 2006, http://cipbook.infracritical.com/book3/chapter2/ch2ref2a.pdf.

[12] Duflos, S., *et al.*, *List of Available and Suitable Simulation Components*, Integrated Risk Reduction of Information-Based Infrastructure Systems (IRRIIS) project, Deliverable D1.3.2, 2006, http://193.175.164.67/?lang=en&nav=241&object=110&item=8786.

[13] Dekker, A.H. & Colbert, B., Scale-free networks and robustness of critical infrastructure networks, *Proceedings of the 7th Asia-Pacific Conference on Complex Systems*, Complex 2004, Cairns, Australia, December 6–10, 2004.

[14] Lee, E.E., Mitchell J.E. & Wallace, W.A., Assessing vulnerability of proposed designs for interdependent infrastructure systems. *Proceedings of the 37th IEEE Annual Hawaii International Conference on System Sciences (HICSS '04)* – Track 2, Big Island, Hawaii, January 05–08, 2004.

[15] Nozick, L.K., Turnquist, M.A., Jones, D.A., Davis, J.R., & Lawton, C.R., Assessing the performance of interdependent infrastructures and optimizing investments. *Proceedings of the 37th IEEE Annual Hawaii international Conference on System Sciences (HICSS '04)* – Track 2, Big Island, Hawaii, January 05–08, vol. 2, 2004.

[16] Svendsen, N.K. & Wolthusen, S.D., Connectivity models of interdependency in mixed-type critical infrastructure networks. *Information Security Technical Report*, vol. 12, no. 1, pp. 44–55, 2007.

[17] Gursesli, O. & Desrochers, A.A., Modelling infrastructure interdependencies using Petri nets. *IEEE International Conference on Systems, Man and Cybernetics*, October 5–8, vol. 2, pp. 1506–1512, 2003.

[18] Krings, A. & Oman, P., A simple GSPN for modelling common mode failures in critical infrastructures. *Proceedings of the 36th IEEE Annual Hawaii International Conference on System Sciences (HICSS '03)* – Track 9, Big Island, Hawaii, January 6–9, vol. 9, 2003.

[19] Chen-Ching, L., Chee-Wooi, T. & Govindarasu, M., Cybersecurity of SCADA systems: Vulnerability assessment and mitigation. *Power Systems Conference and Exposition (PSCE '09)*, IEEE/PES, Seattle, Washington, USA, March 15–18, pp. 1–3, 2009.

[20] Beccuti, M., *et al.*, Quantification of dependencies in electrical and information infrastructures: The CRUTIAL approach. *4th International Conference on Critical Infrastructures (CRIS)*, Linköping, Sweden, April 28–30, pp. 1–8, 2009.

[21] Lu, N., Chow, J.H. & Desrochers, A.A., A multi-layer Petri net model for deregulated electric power systems. *Proceedings of the American Control Conference*, Anchorage, Alaska, USA, May 8–10, vol. 1, pp. 513–518, 2002.

[22] Dudenhoeffer, D.D., Permann, M.R. & Manic, M., CIMS: A framework for infrastructure interdependency modelling and analysis, *Proceedings of the 2006 Winter Simulation Conference*, Monterey, CA, December 3–6, pp. 478–485, 2006.

[23] Tolone, W.J., *et al.*, Critical infrastructure integration modelling and simulation. *Symposium on Intelligence and Security Informatics*, Tucson, AZ, June 10–11, vol. 3073, pp. 214–225, 2004.

[24] Panzieri, S., Setola, R. & Ulivi, G., An agent-based simulator for critical interdependent infrastructures, *Proceedings of the Conference on Securing Critical Infrastructures*, Grenoble, France, October 25–27, 2004.

[25] Casalicchio, E., Galli, E. & Tucci, S. Macro and micro agent-based modelling and simulation of critical infrastructures, *Complexity in Engineering*, Rome, Italy, February 22–24, pp. 79–81, 2010.

[26] Cardellini, V., Casalicchio, E. & Galli, E., Agent-based modelling of interdependencies in critical infrastructures through UML. *Proceedings of the 2007 Spring Simulation Multiconference – Volume 2*. Norfolk, VA, March 25–29, pp. 119–126, 2007.
[27] Repast Agent Simulation Toolkit (http://repast.sourceforge.net/).
[28] Sanders, W.H. & Meyer, J.F., Stochastic Activity Networks: Formal definitions and concepts. *Lectures on Formal Methods and Performance Analysis*, Brinksma, E., Hermanns, H. & Katoen, J.P. (eds), LNCS, Springer Verlag, New York, pp. 315–343, vol. 2090, 2001.
[29] PRIN, Programmi di ricerca scientifica di rilevante interesse nazionale – Progetto di ricerca DOTS-LCCI: Dependable off-the-shelf based middleware systems for Large-Scale Complex Critical Infrastructures, 2008, http://dots-lcci.prin.dis.unina.it/.
[30] IST-FP6-027513 CRUTIAL – CRitical UTility InfrastructurAL resilience (http://crutial.erse-web.it/default.asp).
[31] Chiaradonna, S., Di Giandomenico, F. & Lollini, P., Evaluation of critical infrastructures: Challenges and viable approaches. *Architecting Dependable Systems V*, Lemos, R., Di Giandomenico, F., Gacek, C., Muccini, H., Vieira, M. (eds), LNCS, Springer, Heidelberg, pp. 52–77, vol. 5135, 2008.
[32] Chiaradonna, S., Di Giandomenico, F. & Lollini, P., Interdependency analysis in electric power systems. *Proceedings of the 3rd International Workshop on Critical Information Infrastructures Security (CRITIS 2008)*, Setola, R. & Geretshuber, S. (eds), LNCS, Springer, Berlin/Heidelberg, vol. 5508, pp. 60–71, 2009.
[33] IST-FP6-26979 HIDENETS – HIghly DEpendable ip-based NETworks and Services (http://www.hidenets.aau.dk/).
[34] Bondavalli, A., Lollini, P. & Montecchi, L., Analysis of user perceived QoS in ubiquitous UMTS environments subject to faults. *Software Technologies for Embedded and Ubiquitous Systems*, LNCS, Springer, Berlin/Heidelberg, vol. 5287, pp. 186–197, 2008.
[35] Bondavalli, A., Lollini, P. & Montecchi, L., QoS perceived by users of ubiquitous UMTS: Compositional models and thorough analysis. *Journal of Software*, special issue on Selected Papers of the 6th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS 2008), vol. 4, no. 7, pp. 675–685, 2009.
[36] Sanders, W.H. & Meyer, J.F., Reduced base model construction methods for stochastic activity networks. *IEEE Journal on Selected Areas in Communications*, special issue on Computer-Aided Modelling, Analysis, and Design of Communication Networks, vol. 9, no. 1, pp. 25–36, 1991.
[37] Derisavi, S., Kemper, P. & Sanders, W.H., Symbolic state-space exploration and numerical analysis of state-sharing composed models. *Proceedings of the 4th International Conference on the Numerical Solution of Markov Chains (NSMC '03)*, Urbana, IL, September 3–5, pp. 167–189, 2003.