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

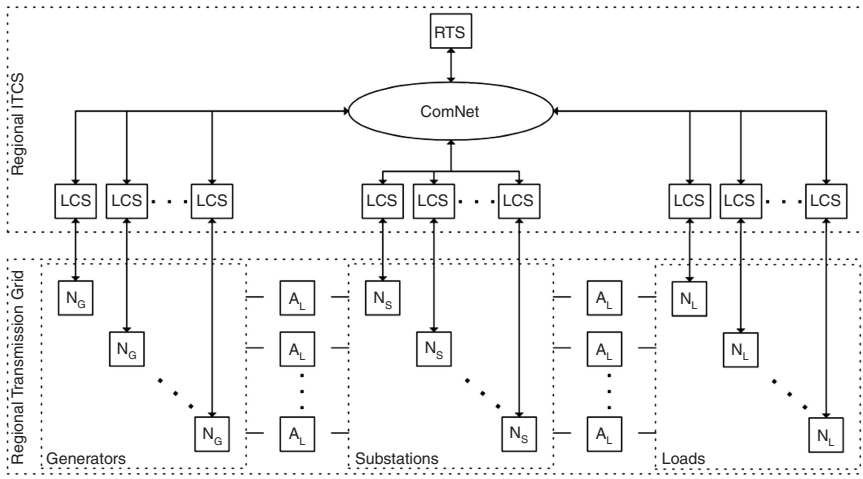

Figure 1: Logical structure of a regional transmission grid, with the associated information control system.

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.

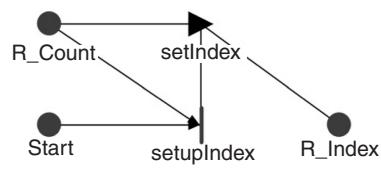

Figure 2: SAN plugin for the indexing of replicas.

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

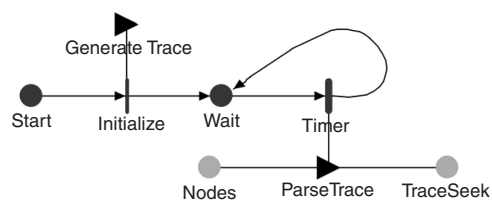

Figure 3: The TraceParser atomic model, which performs the parsing of an external trace.

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.