Reliability Engineering and System Safety 125 (2014) 82–91

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

A modeling framework for the resilience analysis of networked systems-of-systems based on functional dependencies

Roberto Filippini [a,*], Andrés Silva [b]

[a] Independent Research Consultant, 54100 Massa, Italy
[b] GIB Research Group, Facultad de Informática, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Available online 2 October 2013

Keywords:
System analysis
Resilience
Critical infrastructures
Systems-of-systems

ABSTRACT

Critical infrastructures provide services that are essential for the support of everyday activities in modern societies. Being the result of a continuous process of integration of diverse technologies and organizations, they require a multi-disciplinary, systemic approach in order to be understood. In this respect, one of the most challenging issues is the analysis of infrastructures under disturbance or malfunctioning, and their ability to resist, react and recover, in a word the resilience. This paper presents a methodology of resilience analysis of systems of systems, with infrastructures as a special instance. A conceptual representation of the infrastructure, based on the functional relationships among its components, is given and then analyzed with respect to its structural and dynamic properties. Most critical and vulnerable components are identified. The response of the system to failure propagation is simulated in order to check if it is able to cope with them and recover in a resilient fashion. The analysis outcomes are used for a resilience-informed review of the infrastructure.

1. Introduction

Research on critical infrastructures (CI) is of great importance for the security and safety of modern societies and the protection of their strategic assets. The US Department of Homeland Security issued a directive for the identification and protection of CI [1], and a similar one was proposed by the European Council [2]. Both initiatives give emphasis to the development of technical support to policy making, and analysis tools for the assessment of critical infrastructures. As a general recommendation, these analysis tools should privilege a systemic view of the overall infrastructure, in order that the single parts (systems, operators and stakeholders) may recognize themselves and trade-off the different, sometimes conflicting, objectives. A common approach to the identification of measures for risk reduction is also recommended.

Developing such a technical-analytical support is not easy. Modern infrastructures escape conventional definitions of system, and therefore barely fit in any of the existing analysis frameworks. Several arguments exist in support of this thesis. An infrastructure is not conceived as a unique static entity, but it is rather the aggregation of components which adjust their interrelationships in accordance with changes in the operation scenario. The word "rafting" was used to describe this aggregation process, which strongly relies on information and communication technology as the glue that virtually brings the different systems together [3]. In most recent literature, infrastructures are classified as a special instance of systems-of-systems (SoS) of which they possess a number of distinctive features such as operational and managerial independence, geographical distribution, emergent behavior and evolutionary development [4,5]. Clearly, to embrace the broad scope of systems-of-systems is a tremendous challenge. Contributions on this topic are several, see for example [6]. The works of [7,8], inspired to the High Level Architecture standard [9], are likely the most comprehensive in scope and objectives: they propose the integration of different analysis tools into a multi-simulation platform. All the other approaches are more or less specialized and can be hardly stretched out of the scope and the objectives for which they were devised. For example, every analysis framework consider dependencies, as they are the means through which failures propagate and may jeopardize the whole network [10]. Nonetheless, they give emphasis only to some of these, e.g. physical, cyber, geographical, or functional dependencies [11]. The same holds for the choice of model, which is often developed for a specific sector of reference, e.g. gas pipeline, communication networks and power grids [12]. In these cases, the closer focus to specific phenomena implies the impossibility of analysing conflicting goals among different sectors [13,14,6]. The objective of the analysis may also be very diverse, either privileging the assessment of risks [15–19] or the resilience [20,21], but rarely both of them. In view of the above, one may conclude that

* Corresponding author. Tel.: +39 3394129570.
E-mail addresses: rob.filippini@tiscali.it (R. Filippini), asilva@fi.upm.es (A. Silva).

0951-8320/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2013.09.010

the analysis of systems of systems demands an interdisciplinary mindset, as prerequisite for understanding their functioning, but this is not sufficient, and a thorough rethinking out of the box of the problem is necessary [22,23,3].

This paper presents a methodology for the modeling and analysis of systems-of-systems. The methodology is of systemic nature and focuses on functional relationships among system components. A modeling language has been conceived to this purpose [24,25]. Functional dependencies are identified among system components and services, and they are arranged in a dependency network, which is the model of reference for the analysis. The analysis are of two types: structural and dynamic. Structural analysis deals with the way components relate to each other and returns metrics of criticality, vulnerability and interdependency. Dynamic analysis deals with the ability of the network to resist to disturbances by internal buffering, and recover from failure, i.e. its resilience. The analysis reproduces the system response as the sequence of failure and recovery events, from the initial disturbance to the final state. Many system responses may be generated by the same disturbance, depending on the resilience measures in place and their variability. They are labeled as recoverable, if the initial conditions are restored, or vice versa they are accident scenarios. The numerical simulation of the most critical accident scenarios makes it possible to verify whether the network is resilient to them, to which extent, and it also provides recommendations for possible improvements. The methodology is applied to a case study, which is used as proof of concept along the paper.

The paper consists of seven sections. Section 2 presents the overview of the modeling framework and introduction to the modeling language. The structural analysis of the dependency network is in Section 3. Qualitative and quantitative resilience analysis are in Sections 4 and 5 respectively. Section 6 presents some final remarks and a research outlook, while Section 7 will conclude the paper.

2. The modeling and analysis framework

2.1. Overview

The proposed methodology encompasses several steps, from the building of a conceptual model of the system to its transformation into a form amenable to analysis. These are: (1) system representation, (2) structural analysis, (3) qualitative resilience analysis and (4) quantitative resilience analysis. Each of these steps provides the input to the next one. All together they constitute a standalone modeling and analysis framework, shown in Fig. 1. An overview is presented in this section and more details will be given in the following sections.

System representation: The system representation is done in two steps. First the system-of-systems is modeled with its components by means of a suitable graphical language, the Infrastructure Resilience-Oriented Modeling Language or IRML [25]. IRML facilitates the identification of functional relationships among components. They are of three types: (i) provider-user, (ii) producer-consumer, and (iii) controller-controlled. The IRML model is transformed into a dependency network, in which the specificity of every functional relationship is resolved in favor of a simpler, neutral representation. The dependency network is a directed graph with nodes and links accounting for system components and the functional dependencies respectively. The dependency network is the input to the next stage: the structural analysis.

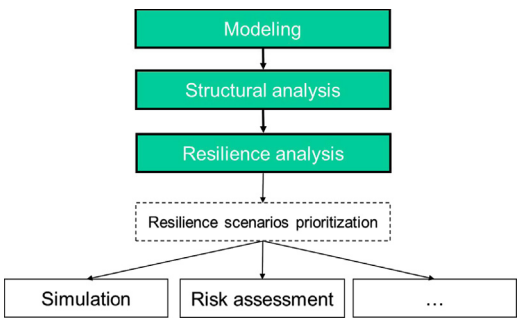

Fig. 1. The modeling and analysis framework.

Structural analysis: The objective of the structural analysis is to return the criticality, vulnerability and interdependency of every node in the dependency network. Coupling and the interaction coefficients are also calculated and they inform on how tight nodes depend on each other.

Qualitative resilience analysis: Qualitative resilience analysis is similar to model checking: all possible system responses to a disturbance applied to a given node are generated. In order to perform the analysis, every node of the dependency network is given two resilience measures, buffering and recovery. Buffering is defined as the ability to retard a disturbance propagation, while recovery is the ability to restore the node functionality after failure. Qualitative analysis performs a pre-screening of all accident scenarios, based on the possibility that they may occur. Among these scenarios, those that lead to a state for which the network cannot recover back to the initial conditions are identified.

Quantitative resilience analysis: Quantitative analysis calculates the system response to a disturbance of a given duration, and for a settings of the network parameters, i.e. times for buffering and times for recovery for every node in the network. The trajectory in the state space is analyzed along the transient period, from the application of the disturbance to the restoration of the initial conditions in the network. The outcome of the analysis is the ability of responding to diverse failure pathologies, in a resilient fashion. The resilience margins are also calculated in order to return the situational awareness of the network, which accounts for the variability of the analyzed scenario. In conclusion, recommendations for the apportioning of resources that are necessary to improve resilience are given.

Risk assessment: In a more general problem formulation, uncertainty can be associated to the resilience measures. Resilience will turn to be the likelihood that the system will recover or not, at an applied disturbance. If costs for service disruptions are estimated, then it is possible to combine consequences and the likelihood of the scenario into an overall figure of risk.

2.2. Introduction to the IRML modeling language

IRML is a graphical language for modeling, analysis and documentation. It was conceived to represent heterogeneous systems that participate in a complex networked infrastructure, and more generally a system-of-systems. One of the features of this modeling language is that of being independent of technological domains. Different technologies and sectors that take part in modern infrastructures can be represented. Clearly, the choice for abstraction limits the level of detail of the representation.

The language is aimed at building models that are not interpreted or compiled as it is the case of a programming language. The context is here different and it is important to disambiguate. In the field of Software Engineering, conceptual modeling has been used for decades [26,27] in order to provide an understanding of complexity. The literature on Ubiquitous Languages [30] is one of the most recent developments in support of this approach, as it

emerges from the need of helping to fill the communication gap between different stakeholders that take part in a software development effort. Conceptual modeling makes it possible to find a balance between detailed descriptions of reality and the much needed abstraction to manage complexity. Of course, conceptual modeling does not exclude that part of it may be supported by software, for instance in order to help the analyst in managing complex models and checking their correctness. From this point of view, the use of IRML is very similar to existing modeling formalisms like entity-relationship diagrams, data flow diagrams, flowcharts, and many others that, for example, are available in UML [28,29]. The main difference between those approaches and IRML is that the latter is analysis-oriented, not suitable for design.

A short introduction of the IRML components and its rules for building a model are given. A full account of the topic can be found in [25]. An IRML model has two types of components: services and domains. The word *domain* refers to phenomena that are usefully treated and represented as a cohesive unit in problem analysis [21]. A domain can be active (agent-domains) or passive (resources). Domains combine together (mainly through control-controlled relationships) in order to constitute a system, which in its turn provides a service.

The IRML components are arranged together according to the following relationships:

- *Provider/user*: A system provides service(s) to other systems.
- *Producer/consumer*: A resource produces a quantity for systems or domains.
- *Controller/controlled*: A domain may control another domain or a resource.
- *Inter-service relationships*: A service may depend on other services.

These relationships determine functional dependencies, and one important feature of IRML is that of ideally "bridging" the descriptive representation of an infrastructure into a network of functional dependencies.

Fig. 2(a) shows a (simplified) IRML representation of a power grid. The model focuses on the dependencies among the production of electricity, transmission, distribution, also including control and communication systems. Every system is described with its constituent domains. Services are exchanged at the system's interface, e.g. communication provides data link to the system control. The IRML model is transformed in the dependency network of Fig. 2(b). The transformation is done by identifying the functional relationships among components, i.e. the nodes of the graph, with their associated goal, i.e. the desired function to perform. The result is a directed graph, in which the hierarchy (higher or lower level of description) is not relevant any more. The six nodes represent the gas network (1), the power plant (2), the control and supervision (3), the transmission (4), the distribution (5) and the communication (6). Seven arcs account for functional dependencies among nodes. Input arcs express the dependency of a node with respect to its ancestors, while output arcs express the dependencies of the descendant nodes with respect to the ancestor node. Forks, junctions and loops are important elements of the topology. A fork means that more nodes will depend on the ancestor, which is the case of nodes 3 (control) and 4 (transmission) with respect to node 2 (power plant). A junction means that the node will depend on more ancestor nodes, which is the case of nodes 3 and 4. Loops are special topologies in which a chain of dependencies closes on itself, which is the case of nodes 3–6. Loops are of particular interest for structural and dynamic implications, and they will be examined in the following section.

3. Structural analysis

A system-of-systems can be given a topology that accounts for the static representation of its components and the way they interact and cooperate. In order to negotiate complexity, this representation requires an adequate modeling abstraction. The idea is to focus on the component's interface, where data, services and quantities are exchanged through functional relationships, i.e. functional dependencies. The following definitions hold.

Functional dependency: A functional dependency is a relationship between two nodes, in which there is an exchange of quantities, data or service, from one node to the other. The functional dependency expresses a necessary condition for the dependent node to perform and function.

Dependency network: The dependency network is the overall representation of all the relevant functional dependencies. It is sector neutral, and its components do not necessarily have to share the same physical domain.

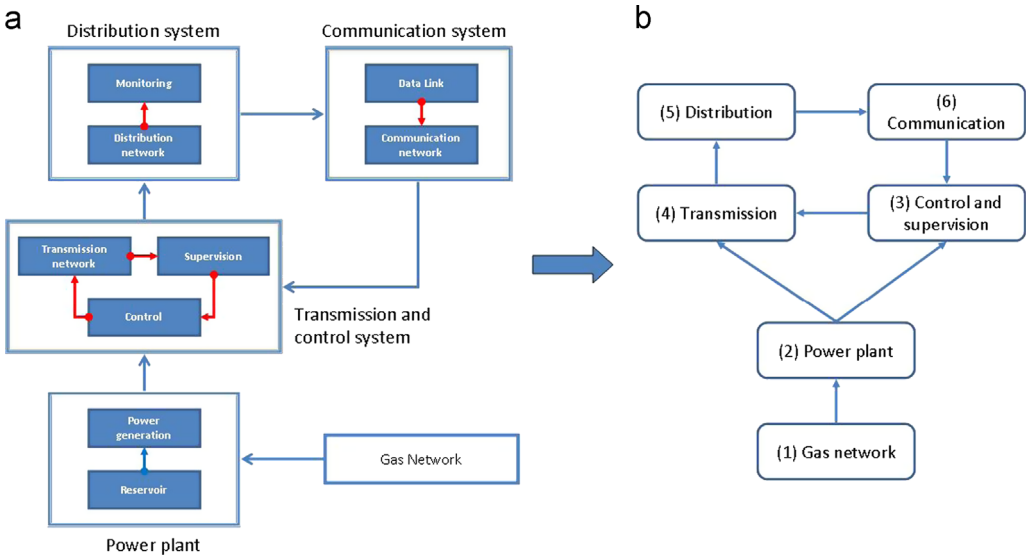

Fig. 2. From the IRML representation of the SoS to the dependency network. (a) IRML model and (b) Dependency network.

The result is a directed graph $G(N,A)$, where N is the set of nodes and A is the set of arcs.

The transformation of the IRML model in Fig. 2(a) into a dependency network of Fig. 2(b) is not necessarily isomorphic, but it responds to the analysis focus, so that more emphasis can be given to certain relationships and less to others. The dependency network is the model of reference for the structural analysis.

3.1. Criticality, vulnerability and interdependency

Criticality, vulnerability, and the interdependency are structural properties that can be analyzed in the dependency network. The following definitions hold.

Criticality: A node k is defined critical with respect to nodes that depend on it, directly or indirectly. These nodes belong to the criticality set of node k, $C(k)$.

Vulnerability: A node k is defined vulnerable with respect to the nodes on which it depends. These nodes belong to the vulnerability set of node k, $V(k)$.

Interdependency: Two nodes k and h are interdependent if they are critical and vulnerable to each other. An interdependency set of a node k includes those nodes that are both critical and vulnerable to k, that is $I(k) = C(k) \cap V(k)$.

Fig. 3 shows the criticality and vulnerability sets of node 2 (power plant). This node is more critical than vulnerable. Its criticality set $C(2)=\{3, 4, 5, 6\}$ is bigger than the vulnerability set $V(2)$, which includes only node 1, the gas supply network. In the example, the failure of the power plant (2) affects all nodes that transmit, distribute, control and consume electricity, either directly or indirectly.

The majority of interdependency relationships are indirect and mediated by other nodes. A consequence of interdependency is that the criticality and the vulnerability sets of a node are not disjoint. In the case of node 2, the intersection of the two sets is an empty set, which means that this node is not interdependent with other nodes. On the contrary, the criticality and vulnerability sets are not disjoint for node 3, see Fig. 4. The interdependency set includes nodes 3–6, which together form the loop.

Structural analysis is completed with two other metrics, the interaction and coupling coefficients. The *interaction* coefficient is proportional to the number of loops that involve a given node: the more the loops, the higher the interaction coefficient. The maximum interaction coefficient corresponds to a graph that is totally connected, i.e. all nodes are directly interdependent. At the lower end, there is the tree like structure, which has no loops. In the example, there is only one loop with 4 nodes involved, and 2 nodes with no interactions. The *coupling* coefficient is calculated on the criticality and vulnerability sets. The coupling is maximum for a star configuration, with all nodes that depend directly on the ancestor. At the lower end, there is the linear chain, which has the lowest coupling. More in general, given a set of n nodes, the coupling will be between $2/(n+1)$, in case of a chain layout, and 1 for a star layout. Fig. 5 shows the coupling coefficients calculated for the criticality and vulnerability of the six nodes. Node 2 is the most critical one, while nodes 3 and 4 result to be the most vulnerable.

The introduced metrics are adapted from Perrow's concepts for the analysis of complex systems [16]. These metrics provide heuristics for classifying networks by their structural properties. High interaction and coupling coefficient unveil the proneness of a network to spread disturbances, which is detrimental to resilience. In contrast, low interaction and coupling coefficients are a more desirable situation for the network. Of course, a structure reflects the particular arrangement of nodes and dependencies that are necessary in order to generate the desired behavior. In principle, another structure with less interaction and coupling might generate the same behavior. These equivalent networks may be compared with their interaction and coupling coefficients. If the network functionality is preserved, then the reduction of the interaction and coupling is empirically related to the improvement of resilience.

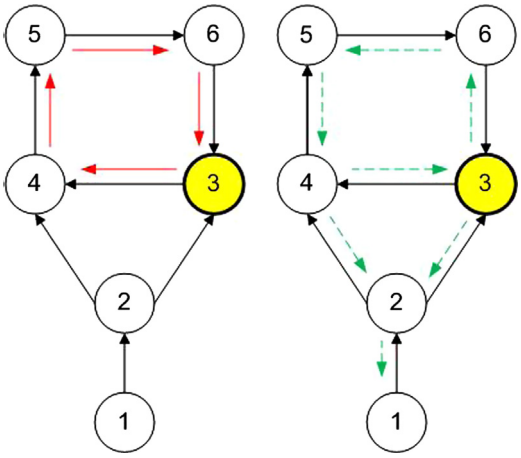

Fig. 4. Criticality and vulnerability sets of node 3.

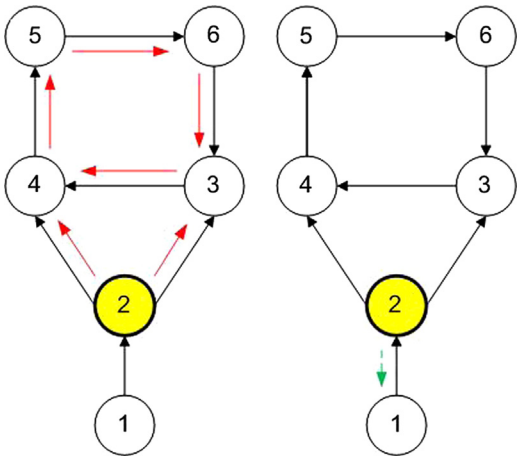

Fig. 3. Criticality and vulnerability sets of node 2.

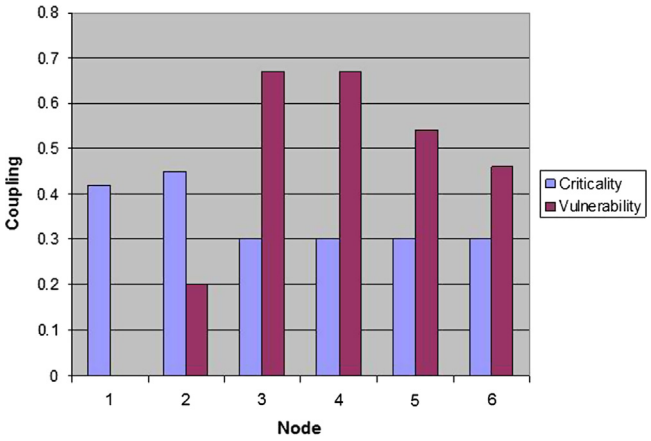

Fig. 5. Criticality and vulnerability coupling for the dependency graph.

4. Qualitative resilience analysis

4.1. The dynamic model

The criticality set $C(k)$ informs about the number of nodes that may be reached by propagation of a disturbance in node k, on the basis of a static representation of the functional relationships. Actually, each node will have defences in place to resist the disturbance (i.e. buffering), or recover back to the initial state, in the case of failure. The inclusion of these features will turn the dependency network into a dynamic model. The following assumptions define the mechanisms that govern disturbance/failure propagation and the recovery throughout the dependency network.

- A1: a disturbance affects one particular node and, from that node, it propagates to its closest descendants in the criticality set;
- A2: a node may fail (after a time to failure T_F), if it is affected by a persistent disturbance from at least one of the ancestor nodes;
- A3: a failed node may recover (after a time to recover T_R) if the input disturbance stopped, which means that all ancestor nodes have recovered in their turn.

The dynamic model for the analysis of resilience is state based, event driven. The state of a node k is a binary variable x_k that takes the values *Up* (1) if the node is functioning and *Down* (0) if it has failed. The initial conditions correspond to every node in its state *Up*. State transitions are governed by the failure and recovery processes. The state transition of a node from *Up* to *Down* is the failure process. It is triggered by the propagation of a disturbance generated in one of the input nodes, which failed in its turn. A disturbance challenges a node to leave the *Up* state to the *Down* state. The state transition is retarded by the activated buffering measures, which allow it to resist for a maximum time interval. This process may be modeled with a *time to failure* T_F or with a degradation rate, depending on the failure and the buffering mechanisms. Both models generalize the concept of *buffering*, or resistance to a disturbance, which can be applied in technical systems and human organizations. The transition from the state *Down* to *Up* is the recovery process. It is triggered by the restoration of the initial conditions at the input nodes. Again, this transition is not immediate. The node activates its recovery measures that allow it to restore the initial conditions. This process may be modeled with a *time to recovery* T_R, or with a recovery rate. Both models generalize the concept of *recovery*.

The dynamic model accounts for a compact set of parameters which are the time to failure (or failure rate) and the time to recovery (or recovery rate) for every node. These are aggregated figures that can be retrieved from operational records of service operators and public utilities, better than the punctual information on the single failure processes. For example, the power plant will have a reservoir that makes it possible to withstand interruptions in the gas supply chain. The operator can estimate its buffering time T_F in a few days. In a similar way the re-activation of the plant, after a gas interruption that consumed the reservoir, can be estimated from the operational experience.

4.2. System response at disturbance

The *system response* is the trajectory in the state space of the dependency network. The sequence of events from the initial disturbance to the end state define the accident scenarios. They are of three types: recoverable, deadlock non-recoverable and time-bounded.

- *Recoverable scenario*: A scenario is recoverable if, whatever the duration of the disturbance, when this stops, it is always possible to recover back to the initial conditions.
- *Deadlock scenario*: A deadlock scenario occurs when all nodes in a loop are all in their *Down* failed state.
- *Time-bounded scenario*: A time-bounded scenario occurs when a recovery deadline of a node exists, at the expiration of which that node(s) cannot be recovered any more.

A *deadlock* is an attractor for the system response and it cannot be removed, even though the disturbance stops. The removal of a deadlock would call for additional resources (e.g. emergency management), which are out-of-the-loop, and not included in the dependency network. Deadlocks occur in systems that for instance produce a quantity and the use a service that depends on the consumption of that quantity, either directly or indirectly. In the given example, the communication depends on the electricity distribution, which depends on the transmission that is operated by the control system. The latter receives data from the communication, which trivially closes the loop.

A *time-bounded* scenario exists for systems that have finite time window to recover. Such systems deal with quantities that may deteriorate or provide non-interruptible services. In the example, there are no explicit time bounded scenarios. Nonetheless, railway transportation could be added and connected (in dependency relationship) to the electricity distribution and the communication systems. The fact that transported goods will deteriorate if not delivered by a certain deadline will make this accident scenario to be time-bounded.

A third type of scenario exists. Failure and recovery events may run one after the other so that the resulting state trajectory will never end up into a final state, i.e. either the initial state or a deadlock. The necessary condition for this scenario to occur is the existence of a loop in combination with particular values of the buffering and recovery measures. Though speculative, these scenarios unveil an unstable behavior of the system response because of its intrinsic complexity, and cannot be excluded a-priori. Again, only an external (out-of-the-loop), coordinated intervention may prevent the occurrence of these scenarios.

4.3. What-if-analysis

Qualitative analysis of resilience is similar to a what-if-analysis. All possible system responses to a disturbance, i.e. accident scenarios, are generated and classified. The model parameters (disturbance duration, T_F and T_R) do not need to be assigned here, being the analysis qualitative.

The generation of accident scenarios is done step-by-step. Assumptions A1–3 supervise this process, which is similar to the construction of an event sequence diagram (ESD) in risk analysis. The main difference is that events are concurrent instead of being mutually exclusive, as *success* and *failure* events in the ESD. Because of that, the branches that depart from a decision block are as many as the number of the active concurrent events. This diagram is called the concurrent ESD. In order to account for event concurrency, each decision block is labeled with two sets of event F and R. They represent respectively the set of active failure events and the set of active recovery events. For an event to be in F or R means that the respective node is in the process of failing or recovering. The occurred failures and/or recovery events are associated to the arc that links one block to the next one. The concurrent ESD is built for a disturbance applied to node k, i.e. the initiating event. As long as the disturbance persists, only failure events are active in the network. When the disturbance stops, recovery events are enabled too (Assumption A3). The rules for

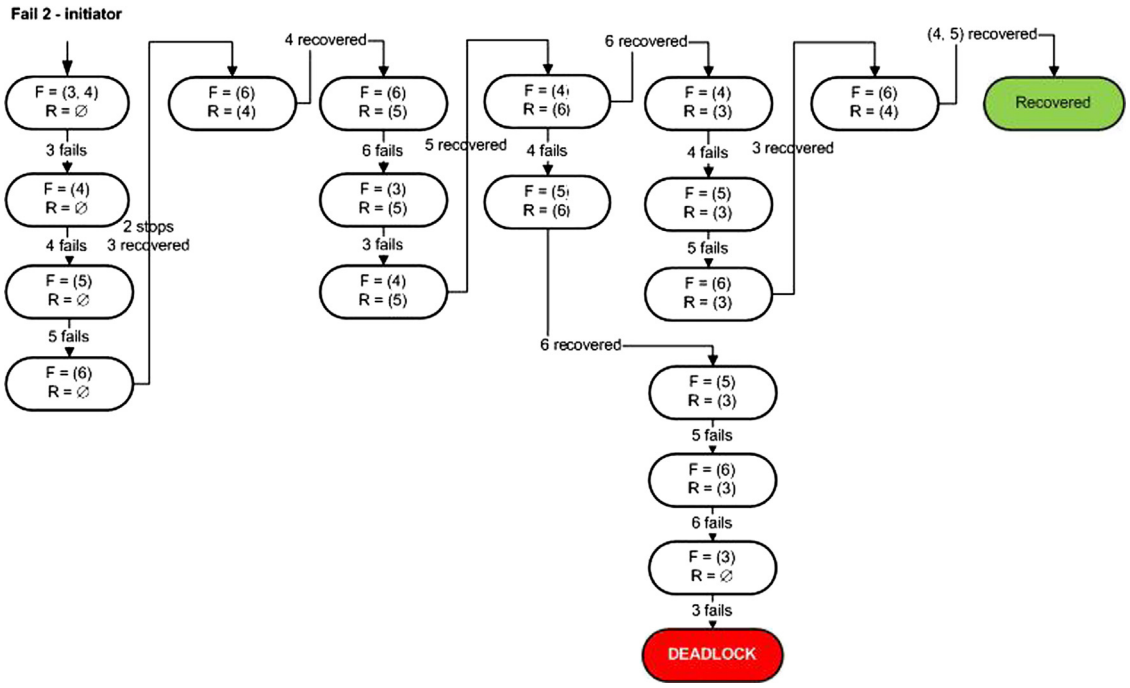

Fig. 6. Two possible scenarios: deadlock and recovered.

selecting next event from the list of the active concurrent events are the following:

- Next event is a failure event: (1) The event is removed from F, (2) a new block is generated and (3) F is updated with the new nodes that are affected by disturbance propagation.
- Next event is a recovery event: (1) This is removed from R, (2) a new block is generated and (3) R is updated with the new nodes that are enabled to recover.

The sequence terminates if R and F are empty, or a deadlock is reached.

An example is shown in Fig. 6. The layout of the concurrent ESD obeys to the following graphical convention: the diagram develops downwards, if a failure occurs, and rightwards into a new column if a recovery occurs. The diagram starts with the failure of node 2 (power plant), i.e. the initiating event, which triggers disturbances at the same time in nodes 3 (control) and 4 (transmission), i.e F=(3,4) and R=empty. The failure propagates up to node 5 (distribution) before the disturbance from node 2 stops. After this event, node 3 recovers and enables the recovery of node 4 too. The sequence of events continues with the failure of nodes 6 (communication) and 3 (control) followed by the recovery of 5 (distribution), up to the point in which failure in 4 (transmission) and recovery in 6 (communication) are concurrent active events. From this point onwards, the diagram takes two different directions. The early recovery of the communication network will result into a recoverable scenario, while the early failure of the transmission will lead to a deadlock. This result is not necessarily the actual network behavior, but just one of the many scenarios that could have generated. Indeed, almost every decision block in Fig. 6 has more than one active event (in F and R), from which other sequences could have departed. The existence of diverse scenarios for a given disturbance is the consequence of system variability. Only the quantitative analysis will make it possible to resolve the indeterminism, and check whether the network will end into a deadlock or it will be able to recover, in a resilient fashion.

The outcomes of the qualitative resilience are the event sequences that lead to a deadlock. If the analysis is repeated for all nodes in the dependency network, and for real-size network, then this task will clearly become infeasible. A significant reduction of scenarios is necessary and can be obtained by defining a set of *termination rules* so to cut branches and/or terminate a sequence if this does not need to be developed further. Possible heuristics are the number of nodes involved and the depth of propagation, which are both related to the likelihood of a sequence to occur. For instance, a long event sequence may be unlikely to occur if events are independent. It is also possible to associate consequences to an event sequence, and this is another heuristic. If the estimated consequence are non-acceptable at a certain stage of the scenario, then there will be no need to develop it further. The generation of accident scenarios and their classifications can be assisted by the software.

5. Quantitative resilience analysis

Quantitative resilience analysis consists of simulating the response of the dependency network to an applied disturbance. The analysis requires the assignment of numerical values to times to failure and recovery for every system node. The model is completed with the applied disturbance, which may affect a single node or multiple nodes, for a given time T_D.

5.1. Resilience analysis

Resilience is the ability of the network to resist a disturbance and recover back to the initial state. As such, the network is resilient or not depending on the resilience measures in place and the disturbance(s) that it has to face. A simple metric of network resilience is the sum of the node's state $r(X)=x_1+x_2+\cdots+x_N$. The quantity $r(X)$ is analyzed from the instant the disturbance is applied, up to the recovery of the network to the initial conditions,

or its structural collapse into a deadlock. The following is a list of attributes of interests that characterize the system response:

1. Resistance at the disturbance: the time the network may resist disturbance, from the time this is applied to the first failure of a node, i.e. $r(X)=N$, for $T_D < T_{min}$.
2. Resilience margin: the maximum duration of disturbance, after which the network is not able to recover back to the initial conditions, i.e. $r(X)<N$, for $T_D > T_{max}$.
3. Duration of the transient: the overall time that the network spends out of the initial conditions, T_{off}.
4. Depth of failure propagation: the maximum number of components that fail during the transient response, i.e. max$[r(X)]$.

Resilience can also be analyzed locally, for instance by isolating the system response in the part of the network of interest. Fig. 7 shows the response of the network in the loop (3, 4, 5, 6) at a disturbance applied in node 2. The resilience function $r=x_3+x_4+x_5+x_6$ is calculated for the deadlock scenario and the recovered scenario in Fig. 6. The disturbance in node 2 is assumed to be an abrupt discontinuity; a step function is considered, which goes from 0 to 1 at time $t=5$, for $T_D=2.5$ time units. The response at the deadlock scenario is obtained for buffering times $T_F=1$ and recovery times $T_R=1$ time unit for every node. Time units and the numerical values of buffering and recovery do not refer to a specific problem set-up but serve as proof of concept. The system response shows an interesting trend, during which recovery is attempted several times but without success. At a certain instant, the resilience function becomes zero and all nodes in the loop enter a deadlock. The recovered scenario can be obtained in two ways: (1) by halving the time to recovery of the node communication (6) or (2) by doubling the buffering time in the same node. The intervention on the recovery time facilitates the restoration to service of node communication before the transmission node may fail, thus enabling earlier the recovery of the node control. In this scenario, the resilience function of the loop recovers back to the initial value ($r=4$) after a transient T_{off} of about 2.4 time units. The second intervention, i.e. doubling the buffering, prevents the failure of node communication, which stops further propagation. Both interventions are successful and prevent the network from falling into the deadlock. Nonetheless, the second solution results in a shorter transient, and also avoids further failure propagation, which instead characterizes the other solution. In conclusion, the result of the analysis suggests investing additional resources in the buffering measure, which guarantees better resilience.

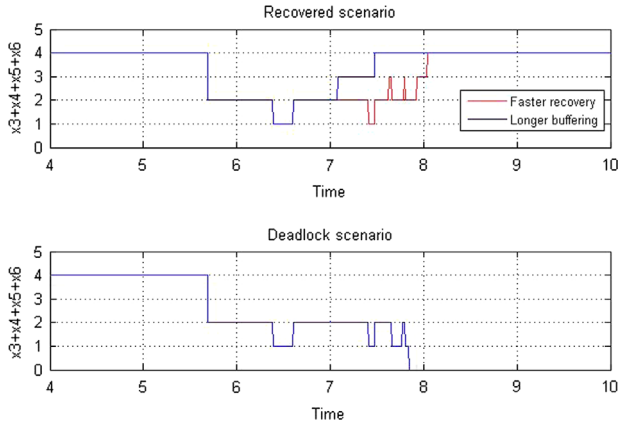

Fig. 7. Simulation of the system response at disturbance within the loop (3, 4, 5, 6).

5.2. Sensitivity analysis and situational awareness

The example of Fig. 7 shows how to intervene effectively in a single node, in order to resolve a deadlock scenario. Nonetheless, even in the favorable situation in which the network withstands and manages to recover, it can still be sensitive to the variability of the applied disturbance and its related resilience measures. Unknowns and uncertainties increase system variability [20]. A consequence of system variability is that the boundary between behavior and misbehavior is blurry and can be hardly determined beforehand. For instance, the response of a SoS may evolve into accident scenarios that, instead of a being caused by a fault, are caused by a legitimate control action that unexpectedly triggered an unstable behavior. Being prepared to the unexpected is another keyword of resilience. The concept of variability is related to the existence of resilience margins of the network, and more in general to the *situational awareness*. Situational awareness is very important in the prevention of accident scenarios; it prepares the operator about what to do next, if a certain scenario will not evolve as expected. In order to calculate the margin of resilience of the network, the duration of the disturbance is increased up to the structural collapse into a deadlock. In the example, the system response is analyzed for a disturbance in node 2, which varies from 1.5 to 3 time units. Four system responses are plotted in Fig. 8. The network is able to restore the initial conditions for a

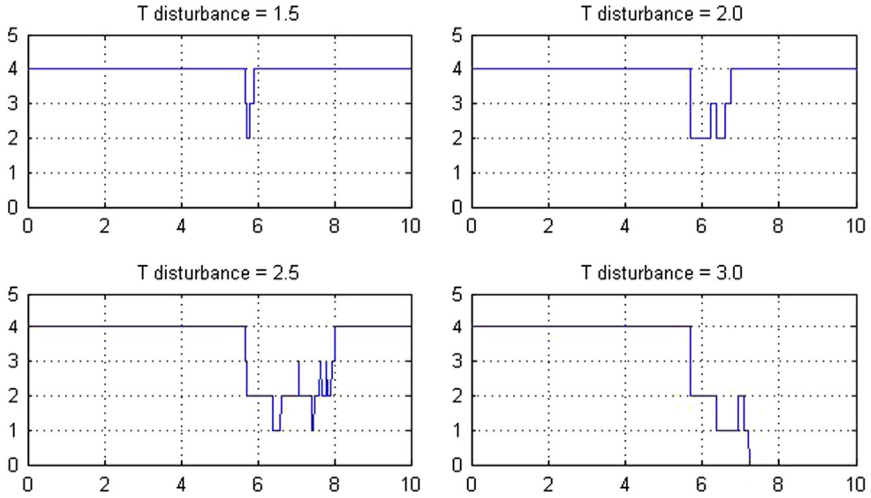

Fig. 8. Response to disturbances of different duration.

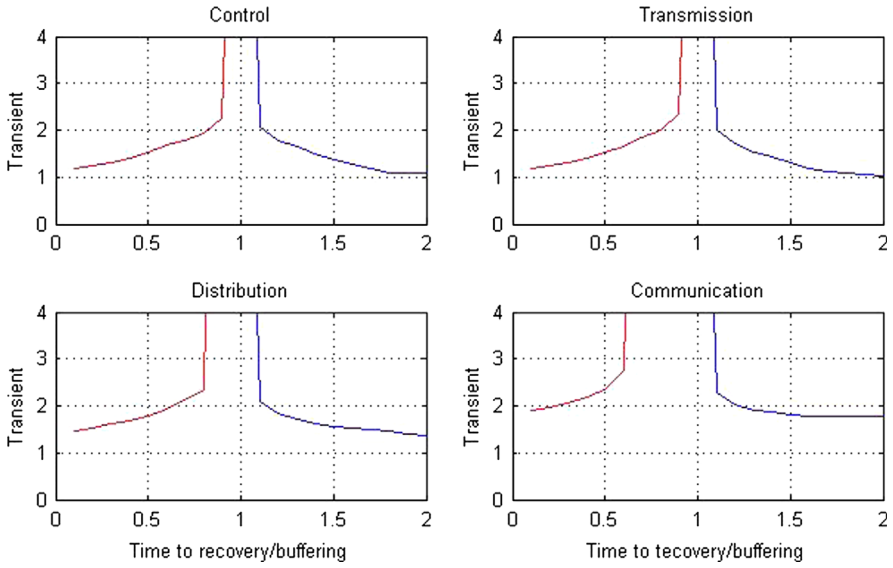

Fig. 9. Duration of the transient response as a function of the recovery (left curve) and buffering times (right curve) for every node.

disturbance that does not exceed (T_{Max}) 3 time units, while it falls into a deadlock if disturbance is longer than 3 time units. The transient duration increases with the disturbance duration, and the same holds for the depth of propagation.

Sensitivity analysis may also consider the variability of the resilience measures. In this case, situational awareness is focused internally towards the resilience capability of each node, and not externally towards the source of disturbance. Fig. 9 shows the response at a disturbance applied in node 2 as a function of the resilience capability of nodes control, transmission, distribution and communication. The scenario is the same; a disturbance is applied at node 2 for a duration of 2.5 time units, and every node is assigned a buffering time and time to recovery equal to 1. With this parameter settings the network falls into a deadlock. The analysis consists of varying the value of the resilience measures, one at the time, and for every node, up to the point that the scenario gets recovered. The duration of the transient as function of recovery (at the left) and buffering (at the right) of the control node is shown in Fig. 9 (top left). If the time to recovery is shortened it will result that the deadlock is resolved, which happens at T_R=0.9 time units. A similar result is obtained if the buffering is increased, and the deadlock is resolved for T_F=1.1. The analysis is repeated for the node transmission, with almost identical results. The node distribution is less effective in terms of transient duration, and this can be explained by the fact that this system is reached later by failure propagation. Even less effective is the improvement of resiliency measures in the communication node. The break-even is obtained for a much shorter T_R=0.6. The transient duration is here longer than in the other cases. In the end, the choice of intervening by halving the time of recovery in the communication node resulted to be the less effective one. A good heuristic for improving resilience is to intervene on the nodes that are the closest at the source of disturbance.

6. Final remarks and outlook

6.1. A comparative literature review of the methodology

The presented methodology brings an original contribution to the modeling and analysing complex heterogeneous systems. It provides novel features, though in a few aspects it is also similar to some of the existing modeling approaches. In this final remarks, some of these are recalled and compared. One of this is the Functional Resonance Accident Model (FRAM) [31]. Like IRML, FRAM is conceived to address resilience in a systemic, holistic way. Small scale, albeit socio-technically complex, systems are within its scope. FRAM is focused on the identification of conditions that may lead to accident scenarios. This is done by decomposing the area interested by accidents in functions and tasks that are involved, each of these elements being susceptible to variability. In FRAM the correspondence of the physical components with the functions and tasks is not trivial. On the contrary, IRML maintains this correspondence functions-systems, which is lost in the dependency graph, but it is always possible to retrieve in the initial model. For what concerns the functional variability, the presented methodology addresses both internal (resilience measures) and external sources (disturbance) of variability, which together concur to the variability of the system response.

The proposal by Johansson [32] focuses on the modeling and analysis of interdependencies and vulnerabilities in critical infrastructures. These are discovered by removing nodes or links in the network and then analysing the consequences. Most critical components, i.e. those ones that have the largest impact, are identified. The methodology is based on the representation of failure/repair mechanisms. Every system is given a binary state, available or not available. However, in resilience, it is important that nodes may be represented with their inertia to fail and the time to recover, which accounts for intermediate conditions that are in between functioning and failed states and where an acceptable service degradation can still be negotiated. This modeling feature is an added value of our proposal.

Another interesting proposal is that of Utne et al. [19]. The focus is the analysis of the interdependencies related to hazardous events. In IRML, the building of the model of an interconnected network of systems of systems is the input to the subsequent analysis of interdependencies with their resilience implications. In the proposal by Utne et al., instead, a lot of detailed information on a particular event of interest needs to be collected in order to prepare the data set for the analysis; only interdependencies related to that event are analyzed and developed forward in a cascade diagram, similar to an event tree. In this respect IRML analysis is more general and abstract in scope, at the cost of lacking of representation details, while Utne's analysis is more specific, event-focused, at the cost of losing generality. Finally,

there is the High Level Architecture (HLA) standard [9]. This is a very interesting framework, though it stands apart if compared to the others. The HLA provides a platform for the integration of simulators, each one specialized to different domains or phenomena of interest. In this way, it is theoretically possible to build and simulate large scale accident scenarios. A major shortcoming is that complexity cannot be negotiated in the same effective way as in the IRML framework. In HLA all details count in order to make simulations as realistic as possible, while in IRML, the goal is that of providing high level understanding of accident scenarios and resilience, at a reasonable computational effort.

6.2. Research outlook

The presented proposal is well developed in many of its features, but directions for refinement and improvements exist. The following is a list of "things to do", which may represent the future research outlook on the topic.

The language: IRML needs a review of some of its constructing elements. This has to be done without loosing of generality, for better adhering to real applications. For example, an IRML 2.0 can allow modularization. At the time of this proposal this was not considered, but it may turn to be of great help if parts of the network ask for further investigations.

The model of the network response: The model is a discrete event system. Its mathematical formalization was not the focus, at least in this paper. Other approaches would have met the same objectives, such as a Petri Net, or a discrete event simulation language. Actually, one may consider dynamic modeling as still an open issue in this framework. In order to facilitate the presentation of ideas, the system response to disturbance was governed by a deterministic cause-effect dynamics. This may also sound as a limitation in real applications. More generally, model parameters will be given ranges of uncertainty. As a consequence, resilience analysis will return the distribution of the system responses at a disturbance. In this case, a Monte Carlo simulation is the appropriate tool. Again, this deals with modeling aspects that are out of the scope of this paper, but that will certainly concern future research activities.

The resilience analysis: Other specific pathologies can be considered. For example, a sequence of disturbances on the same node will mimic the scenario of a communication network that is not able to manage traffic overloads and crashes. Multiple sources of disturbance that affect different nodes can be also analyzed. This latter scenario would perfectly fit in security domains, where the ability to deal with coordinated attacks is the focus.

Resilience and risk informed design: Design implications were addressed with a few examples on situational awareness, in which the apportioning of additional buffering and recovery measures was informed by the calculation of resilience margins. Nonetheless, this is just one of the quantities one has to look at. The risk associated to an accident scenario counts as well and it has to be taken in account for the review of the resilience measures. For instance, a cost may be associated to the service outage of a node, either in term of safety or production losses, depending on the context.

Software support: The model building lends itself to be assisted by software, as well as the tools for the structural analysis, what-if-analysis, and the simulation of the system responses.

7. Conclusions

This paper presented a methodology for the resilience analysis of systems-of-systems and critical infrastructures as a special instance. Specialization is here relaxed in favor of a systemic view. Functional dependencies among components (e.g. producer–consumer, provider–user, controller–controlled) are the modeling focus, and are represented by a dependency network, which is analyzed with respect to its structural and dynamic properties. The former returns criticality, vulnerability, and other topology related metrics such as coupling and interaction coefficients. Dynamic analysis copes with the network response to a disturbance. This is first done by what-if-analysis, which returns a prescreening of the proneness of the network to develop accident scenarios by failure propagation. The most critical of these scenarios are simulated. If the network is able to to cope with the accident, then it is resilient. On the contrary, a reconsideration of the resilience measures may be necessary. The application of the methodology was exemplified for a case study.

The most important features of the methodology are briefly recalled in these conclusions. The focus on functional dependencies is very important: it broadens the scope of the analysis and makes it possible to develop a conceptually simple model for the analysis. Structural and dynamics quantities can be derived from this model, so to provide a comprehensive picture of the system under study. The system response is driven by concurrent failure and recovery events, and accounts for the progressive degradation of the infrastructure as a function of disturbance(s) and the defences in place. Another important feature is the possibility of performing a sensitivity analysis with respect to system variability, that is the ability of the network to deal with uncertainties in the operation scenarios, thus enforcing situational awareness. While these and other features of the presented framework are well consolidated, developments in some aspects of the methodology are also envisaged, as it was outlined in the previous section. We left a few remarks on resilience itself in this concluding section. The resilience analyzed in this paper is very similar to system stability: a system resists and reacts to a disturbance as long as this is within its capabilities. Nonetheless, in the literature the concept of resilience embraces other aspects which are more difficult to translate into quantities, such as the ability to anticipate events, and to learn from experience [20]. The ability to prevent and anticipate can be integrated as an additional resilience feature of the model, for example by considering another layer of interdependencies among nodes that supervise the state of the network and communicate anomalies before these may propagate. The ability of learning from experience is of great interest but rather speculative, and to deal with this resilience feature is beyond the scope of the proposal.

In conclusion, the methodology with its analysis tool-set provides a standalone framework for assessing the resilience of complex networked systems-of-systems. It may be of support to stakeholders and decision makers, operators of infrastructures, regulation authorities, and government institutions [1,2]. In a more ambitious perspective, the methodology may be integrated in the resilience informed design of systems-of-systems.

References

[1] Department of Homeland Security. Critical infrastructure identification, prioritization, and protection. Homeland Security Presidential Directive 7; December 2003.
[2] European Council. On the identification and designation of European critical infrastructures and the assessment of the need to improve their protection. Official Journal of the European Union. Council Directive 2008/114/EC; 8 December 2008.
[3] Egan MJ. Anticipating future vulnerability: defining characteristics of increasingly critical infrastructure-like systems. Journal of Contingencies and Crisis Management 2007;15(1):4–17.
[4] Sousa-Poza A, Kovacic S, Keating C. System of systems engineering: an emerging multidiscipline. International Journal of System of Systems Engineering 2008;1:1–17.
[5] Jamshidi M, editor. System of systems engineering, innovations for the 21st century. John Wiley & Sons; 2009.

[6] Yusta JM, Correa GJ, Lacal-Arategui R. Methodologies and applications for critical infrastructure protection: state-of-the-art. Energy Policy 2011;39(October (10)):6100–19.
[7] Nan C, Eusgeld I. Adopting HLA standard for interdependency study. Resilience Engineering and System Safety 2011;96(January (1)):149–59.
[8] Eusgeld I, Nan C, Dietz S. System-of-systems approach for interdependent critical infrastructures. Reliability Engineering and System Safety 2011;96(June (6)):679–86.
[9] IEEE 1516. Standard for modeling and simulation high level architecture. IEEE; 2010.
[10] Panzieri S, Setola R. Failure propagation in critical interdependent infrastructures. International Journal in Modeling identification and Control 2008;3 (1):69–78.
[11] Rinaldi SM, Peeremboom JP, Kelly TK. Identifying, understanding and analyzing critical infrastructure interdependencies. IEEE Control Systems Magazine 2001;21(6):11–25.
[12] Bompard E, Napoli R, Xue F. Analysis of structural vulnerabilities in power transmission grids. International Journal of Critical Infrastructure Protection 2009;2:5–12 Elsevier.
[13] Pruyt E, Thissen W. Transition of the European electricity system and system of systems concepts. In: IEEE international conference on systems of systems engineering, SoSE'07; September 2007.
[14] Fritzon A, Ljungkvist K, Boin A, Rhinard M. Protecting Europe's critical infrastructures: problems and prospects. Journal of Contingencies and Crisis Management 2007;15:30–41.
[15] Leveson N. A new accident model for engineering safer systems. Safety Science 2004;42(4):237–70.
[16] Perrow C. Normal accidents: living with high-risk technologies. Princeton University Press; 1999 updated edition.
[17] Sterbenz JPG, Hutchison D, åtinkaya EK, Jabbar A, Rohrer JP, Schoeler M, et al. Resilience and survivability in communication networks: strategies, principles, and survey of disciplines. Computer Networks 2010;54:1245–65.
[18] Kroeger W, Zio E. Vulnerable systems. Springer; 2011.
[19] Utne IB, Hokstad P, Vatn J. A method for risk modeling of interdependencies in critical infrastructures. Reliability Engineering and System Safety 2011;96(June (6)):671–8.
[20] Hollnagel E, Woods DW, Leveson N, editors. Resilience Engineering: Concepts And Precepts. Ashgate; 2006.
[21] Jackson S. Architecting resilient systems. Wiley; 2010.
[22] Valerdi R, et al. A research agenda for system of systems engineering. International Journal of System of Systems Engineering 2008; 1: 171–88. Inderscience Publisher.
[23] Kroeger W. Critical infrastructures at risk: a need for a new conceptual approach and extended analytical tools. Reliability Engineering and System Safety 2008;93:1781–7.
[24] Filippini R, Silva A. Modeling language for the resilience assessment of networked systems of systems. In: Proceeding of the conference ESREL 2011, Troyes; 18–22 September 2011.
[25] Silva A, Filippini R. Infrastructure (resilience-oriented) modeling language: IRML. A proposal for modeling infrastructures and their connections [JRC Scientific and Technical Reports]. JRC63302. JRC of the European Commission; 2011.
[26] Davis A. Software requirements: objects functions and states. 2nd ed. Prentice-Hall; 1993.
[27] Wand Y, Monarchi DE, Parsons J, Woo CC. Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems 1995;15(4).
[28] Davies I, Green P, Rosemann M, Indulska M, Gallo S. How do practitioners use conceptual modeling in practice? Data and Knowledge Engineering 2006;58:358–80.
[29] The unified modelling language. Object management group. http://www.uml.org. Last visited May 5, 2013.
[30] Shore J. The art of agile development. O'Reilly 2007.
[31] Hollnagel E, Goteman O. The functional resonance accident model. In: Proceedings of the cognitive system engineering in process plant 2004, CSEPC 2004. p. 155–61.
[32] Johansson J, Hassel H. An approach for modelling interdependent infrastructures in the context of vulnerability analysis. Reliability Engineering and System Safety 2010;95(12):1335–44.