A process-based dependency risk analysis methodology for critical infrastructures

George Stergiopoulos and
Vasilis Kouktzoglou

Department of Informatics,
Athens University of Economics and Business,
76 Patission Ave., GR-10434, Athens, Greece
Email: geostergiop@aueb.gr
Email: kouktzogloub@aueb.gr

Marianthi Theocharidou*

European Commission,
Joint Research Centre (JRC),
Directorate E.
via E. Fermi, 2749, I-21027, Ispra (VA), Italy
Email: marianthi.theocharidou@ec.europa.eu
*Corresponding author

Dimitris Gritzalis

Department of Informatics,
Athens University of Economics and Business,
76 Patission Ave., GR-10434, Athens, Greece
Email: dgrit@aueb.gr

Abstract: This paper applies research in dependency modelling to a process-based risk assessment methodology suitable for critical infrastructures. The proposed methodology dynamically assesses the evolution of cascading failures over time between assets involved in a business process of an infrastructure. This approach can be applied by a CI operator/owner to explore how a failure in a single component (asset) affects the other assets and relevant business processes. It could also be applied in an analysis that includes multiple CI operators in the same supply chain to explore the dependencies between their assets and explore how these affect the provision of key societal services. The paper presents a proof-of-concept tool, based on business-process risk assessment and graph modelling, and a realistic case example of a rail scheduling process. The approach allows risk assessors and decision makers to analyse and identify critical dependency chains and it can reveal underestimated risks due to dependencies.

Keywords: risk assessment; business process; asset; dependency; cascading failures; risk chains; likelihood; impact; critical infrastructure.

Reference to this paper should be made as follows: Stergiopoulos, G., Kouktzoglou, V., Theocharidou, M. and Gritzalis, D. (2017) 'A process-based dependency risk analysis methodology for critical infrastructures', *Int. J. Critical Infrastructures*, Vol. 13, Nos. 2/3, pp.184–205.

Biographical notes: George Stergiopoulos is a Senior Researcher and IT Security Consultant. He has a PhD in Information Security Software and Critical Infrastructure Protection from the Department of Informatics, Athens University of Economics and Business, Athens, Greece, an MSc in Information Technology from Athens University of Economics and Business, Athens, Greece and a BSc in Computer Science from the University of Piraeus. He is a member of the Information Security and Critical Infrastructure Protection (INFOSEC) Laboratory (http://www.infosec.aueb.gr). He is also an accredited lecturer for the University of Derby. His current research interests focus on critical infrastructure protection, risk assessment, application security and software engineering. His professional experience includes working as a risk assessment consultant in projects using ISO-certified methodologies for developing enterprise security plans, business continuity plans, destruction recovery plans and assessing enterprises against IT threats and risks through governance, compliance, identification, and validation. He also works as an IT security penetration tester.

Vasilis Kouktzoglou holds a BSc in Computer Science from the Athens University of Economics and Business and an MSc in Information Systems from the Athens University of Economics and Business. His current research interests include critical infrastructure protection, risk assessment, information security and social engineering.

Marianthi Theocharidou is a Project Officer of the European Commission's Joint Research Centre. She holds a BSc in Informatics, an MSc in Information Systems and a PhD in ICT Security from Athens University of Business and Economics (Greece). Currently, she is working on the European Reference Network for Critical Infrastructure Protection (ERNCIP) and on the EU-funded H2020 project IMPROVER. Before joining JRC, she was an Adjunct Lecturer and a senior member of the Information Security and Critical Infrastructure Protection Research Group with the Department of Informatics of the Athens University of Economics and Business. Her published work includes more than 35 scientific publications in peer-reviewed journals and conferences, in various topics such as critical infrastructure protection and resilience, risk assessment and information security.

Dimitris Gritzalis is Associate Rector and Professor with the Dept. of Informatics of Athens University of Economics and Business, Greece. He also serves as the Director of the Information Security and Critical Infrastructure Protection (INFOSEC) Laboratory and as the Director of the MSc Programme in Information Systems. He is the Academic Editor of *Computers & Security Journal* (Elsevier). His current research interests include critical infrastructure protection, social media intelligence, information security, and digital forensics.

1 Introduction

Typical risk assessment (RA) for critical infrastructures (CIs) includes dependency analysis when the RA refers to a cross-CI or cross-sector analysis. It is not part of the operator's RA by definition; rather it depends on the sector and the CI's maturity level. Still, relevant research in this area has proved that dependency analysis of CIs can yield interesting results when assessing potential threats. Being an intensive problem when it comes to cross-sectoral, cascading and common-cause failures, few tools and methodologies have been able to automatically analyse these dependencies, map the chain of effect and propose mitigation countermeasures. The problem intensifies when attempting a dynamic, time-based dependency analysis.

This paper utilises a graph-based risk analysis methodology previously proposed for analysing cross-infrastructure dependency failures (Stergiopoulos et al., 2016a; Kotzanikolaou et al., 2013a, 2013b, 2013c) and applies it to a process-based RA. The proposed methodology can dynamically assess the evolution of cascading failures over time between assets involved in an infrastructure's business processes. Various impact growth models are employed to capture slow, linear and rapidly evolving effects, but instead of using static impact ranks, the impact evolution in each asset dependency is modelled by a fuzzy system that also considers the effects of nearby dependencies. For each dependency, this is achieved through the quantification of impact on a time axis in the form of many-valued logic. The methodology is also able to analyse failures triggered by concurrent common cause cascading threats. The proposed methodology for process-based asset dependency analysis was implemented in Java and tested on the IT systems of a real-world CI. The output of the tool can assist decision makers in proactively analysing dynamic and complex dependency risk paths between assets and business processes by identifying potentially underestimated low risk asset dependencies and reclassifying them to a higher risk category or by simulating the effectiveness of countermeasures on assets.

1.1 Motivation

Several RA methodologies and tools have been developed; some focus on the assets (CRAMM, 2010; Caralli et al., 2007; http://www.ar-tools.com/en/tools/pilar/) and others on the business processes (de Haes and Debreceny, 2013; Simonsson et al., 2007). Other methodologies focus on the risk derived from CI dependencies (Stergiopoulos et al., 2016a; Kotzanikolaou et al., 2013a, 2013b, 2013c; Alpcan and Bambos, 2009) and their potential cascading effects. Most methodologies and tools are usually entity-specific and oriented towards providing assessment reports and countermeasures on specific parts of an IT system; either assets or business processes alone. These tools and methodologies are very useful for targeted analyses of scenarios (e.g., identifying the critical assets and processes in an infrastructure). However, they may fall short when high-level analyses are needed in order to model asset and process dependency scenarios that may include asset dependencies from external infrastructures. One example is the identification of dependency paths of assets that simultaneously affect multiple business processes.

The overall impact (or risk) of a given infrastructure failure on a multitude of its business processes is not a tangible value, especially when multi-order asset dependencies are present. A high-level risk analysis of asset dependencies between all processes allows the identification of complex cascade or common-cause risk paths and the comparison of alternative mitigation strategies.

1.2 Contributions

This paper utilises parts of a CI dependency analysis proposed in Stergiopoulos et al. (2016a) and Kotzanikolaou et al. (2013a, 2013b, 2013c) in process-based RA. The new, proposed methodology dynamically assesses the evolution of cascading failures over time between assets involved in the same business processes. This approach can be applied by a single CI operator to explore how a failure in a single component (asset) affects the other assets and relevant business processes. It could also be applied in an analysis that include multiple CI operators in the same supply chain to explore the dependencies between their assets and how they affect key societal services (of this particular supply chain).

This paper also presents a proof-of-concept tool based on business-process RA and graph modelling. Particularly, we developed a proactive modelling and asset dependency analysis tool for evaluating large-scale, cross-sectoral asset dependency scenarios based on business processes that they adhere to. This allows risk assessors and decision makers to analyse and identify critical dependency chains at the preparedness stage. Thus, it can reveal underestimated risks due to dependencies. The methodology can also assess alternative risk mitigation strategies and contribute in enhancing resilience.

Any CI is an instance of a system of systems (SoS) (Jamshidi, 2010; Sousa-Poza et al., 2008; Gorod et al., 2008). While regular complex IT systems still got boundaries and defined architecture, a SoS is blurrier in boundaries and may evolve in time (Giannopoulos et al., 2012). An infrastructure does not work in isolation, so disruptions propagate to entire networks of dependent services. The modelling scope of the presented methodology stems from the concept of SoS (Giannopoulos et al., 2012). The proposed methodology can model systems of systems no matter the concept behind their business process model and can thus support the detection of any type of service failure found in CIs, unlike most typical RA methodologies which tend to model specific IT systems with predefined types of links between standard threats and assets. New types of processes, input or assets can be modelled on-the-fly as nodes without having to update the methodology nor any supporting tools.

2 Literature review

Managing risk effectively protects CIs against threats, reduces their vulnerabilities and potential impacts from threat manifestation (economic, societal or otherwise). Currently, a plethora of heterogeneous RA methods are available, having a different focus based on the type of organisation (e.g., government agency, SME, etc.) or based on the critical sector. Still, no RA methodology exists that can act as a 'silver bullet': Analysts have to choose from a plethora of different methods to select one that best fits the CI to be assessed. According to the work of Giannopoulos et al. (2012), the selection of an appropriate methodology to assess the information security risk of a CI depends on several criteria (scope and objectives of the methodology, applied techniques and standards, interdependencies coverage, etc.).

Several methods currently exist to assess risks in infrastructures and systems. Most of them require different skills and experience from their users. Asset-based methodologies like MAGERIT, CORAS and MEHARI involve their users in the assessment (Amutio et al., 2014; CLUSIF, 2010; CORAS, 2010). CRAMM, OCTAVE and RiskSafe require extensive standardised documentation throughout RA to ensure traceability of results (CRAMM, 2010; Platinum Squared, 2014; Caralli et al., 2007). In addition, each of these RA methods demands a knowledgeable team (analysts, system administrators, users, etc.) with a comprehensive set of skills and experience. On the other hand, there are methods, such as EBIOS, which are tailored to administrators (ANSSI, 2010). The aforementioned methods analyse relationships between assets and threats along with the impact of the occurrence of a threat and the occurrence of the threat in relation to the existing vulnerabilities of the system (e.g., CRAMM, CORAS) (ENISA, 2006).

Still, most modern RA methodologies and tools are ‘asset-based’, meaning that they approach the entire RA process through the value of assets (data, information or resources) of an infrastructure. ISO 27001:2013, an international standard for the certification of IT systems, clearly states the ability to utilise process-based RA: “You do not need to use the assets-threats-vulnerabilities methodology to identify risks - for example, you can identify risks based on your processes, based on your departments, using only threats and not vulnerabilities, or any other methodology you like” ((ISO, 2013)). On top of those standards, the 2016/1148 Directive of the European Parliament and the Council states that the existence of a service-driven risk management is needed (European Parliament, 2016). This implies an approach, which places the primary focus on processes, as the objective is to protect essential for the citizen services. Such services (processes) could be provided by single or multiple infrastructures. The RA method described in this paper adopts such an approach and, also, takes into account the requirements described within ISO 27005 (ISO, 2011) and NIST SP800-39 standards (NIST, 2011).

Regarding the analytical techniques used by the various methods, they are either qualitative, quantitative or their combination. Methods such as EBIOS, MEHARI, CRAMM and OCTAVE, follow the qualitative approach, which relies on expert opinion, but introduces a degree of subjectivity on the results. Methods that follow the quantitative or the hybrid approach, such as MAGERIT and CORAS, allows a mathematical evidence to be used in support of decision making under uncertainty, but they require high-quality input data and a well-developed project model. Our methodology can be considered a hybrid, since it follows a qualitative approach on asset identification, but also supports a mathematical formula to support decision making through fuzzy logic (FL), when multiple business processes exhibit different risk valuations over time. Our approach combines a method for discovering dependency risk paths with an automated modelling and analysis tool. It enables the dependencies-per-business-process of interconnected assets to be depicted as a graph and critical paths to be identified.

For identifying a way to cope with business-processes in our RA in our methodology, we utilised the methodical analysis of COBIT 5. COBIT 5 is a comprehensive framework for developing, implementing, monitoring and enhancing information technology governance and management practices (de Haes and Debreceny, 2013). It offers a business process-based RA method by identifying and modelling end-to-end business and functional areas of responsibility, while taking into consideration the IT-related interests of internal and external stakeholders. However, even if COBIT 5 is able to bridge the gap between business control models and IT asset-based RAs, it comes with the disadvantage of utilising over-complicated concepts and structures that make COBIT difficult and time-consuming to apply it as a RA tool (Simonsson et al., 2007). Our approach tries to simplify the business-process modelling of COBIT by utilising each business process output as a single end-node in a chain of inter-depended assets that are utilised in the specific business process.

According to Ouyang (2014), there exist multiple mathematical models that study dependencies of CIs; these models most often fall into one of the following categories:

1. empirical
2. agent-based
3. system dynamics-based
4. economic theory-based
5. network (topology or flow)-based
6. others (hierarchical holographic modelling-based, high-level architecture, etc.).

Admittedly, a practical comparison of the proposed methodology with multiple RA methods against a real-world CI testing example is not feasible, since this would require multiple access to both numerous proprietary methodologies and tools (i.e., great cost) along with a CI that would allow us to re-assess its risk in its entirety; something that would need gratuitous amounts of time and extra pages in this paper. Still, the advantages of the presented method compared to others using high-level quality criteria is presented at Section 5 ‘Comparison with other approaches’.

3 Building blocks

Two fundamental building blocks are used in the proposed methodology, extracted from previous research (Stergiopoulos et al., 2016a; Kotzanikolaou et al., 2013a, 2013b, 2013c) and redefined to suit our RA needs:

1. the subjacent multi-risk dependency analysis methodology for cascading failures
2. the fuzzy modelling approach applied for the time-based analysis of dependencies.

3.1 Multi-risk asset dependency analysis methodology

Essentially, a business process is a step-by-step description of what users have to do to accomplish a specific task. Those steps utilise resources and assets from the IT system. Mapping these asset dependencies per business process allow us to calculate dependency chains and utilise them to assess the cascading and cumulative risk of potential threats on a business process. While such a detailed analysis may not be required in typical information systems, it could be valuable when analysing complex systems of high societal value.

The multi-risk dependency analysis method (Stergiopoulos et al., 2016a; Kotzanikolaou et al., 2013a, 2013b, 2013c) is a network-based modelling technique that takes advantage of the results of organisation-level RAs carried out by owners and operators of enterprises. Directed graphs are used to visualise the relationships (dependencies) between assets arising from the business processes that manage them in order to assess the risk of n^{th}-order dependencies. A dependency can be defined as a "one-directional reliance of an asset, system, network or collection thereof – within or across sectors – on an input, interaction or other requirement from other sources in order to function properly" (Amutio et al., 2014). In the proposed methodology, a graph is used in order to model the dependencies. Let $G = (N, E)$ denote this graph where N is a set of nodes (assets) and E is a set of edges (or dependencies). The graph is directional in nature to model dependencies from one asset to other assets. An edge from node A_i to node A_j, i.e., $A_i \rightarrow A_j$, implies a risk relation that is derived from the dependence of node A_j on A_i due to a business process relating them. This relation is quantified using the impact $I_{i,j}$ and the likelihood $L_{i,j}$ of a disturbance occurrence. The product of these two values is defined as the dependency risk $R_{i,j}$ to asset A_j due to its dependence on asset A_i. Each edge of the graph is associated with a numerical value which ascribes the level of the cascade-resulting risk for the receiver due to the dependency. A risk scale [1 … .9], where 9 is the most rigorous risk, is used to depict this risk. All the parameters ($L_{i,j}$, $I_{i,j}$, $R_{i,j}$) are defined in order to assess the risk of first-order dependencies. The main input to this method is provided by enterprise owners and operators, and refers to the obvious upstream dependencies as mentioned above.

3.1.1 n^{th}-order dependency risk

Given the first-order dependencies as described in the subsection above and according to Kotzanikolaou et al. (2013a), it is possible to assess the potential n^{th}-order cascading risks using a recursive algorithm. Let $A = (A_1, \ldots, A_m)$ be the set of assets. Let $A_{Y_0} \rightarrow A_{Y_1} \rightarrow \ldots \rightarrow A_{Y_n}$ denote a chain of connected assets of length n derived from business processes. Then, the recursive algorithm examines each of these nodes as the potential root of a cascading effect (denoted as A_{Y_0}) and computes the dependency risk DR exhibited by A_{Y_n}, due to the nth-order dependence.

If $A_{Y_0} \rightarrow A_{Y_1} \rightarrow \ldots \rightarrow A_{Y_n}$ is a chain of dependencies, $L_{Y_0,\ldots,Y_n}$ is the likelihood of the nth-order cascading effect and $I_{Y_{n-1},\ldots,Y_n}$ is the impact of the $A_{Y_{n-1}} \rightarrow A_{Y_n}$ dependency, then the cascading risk exhibited by A_{Y_n} due to the n^{th}-order dependency is computed as:

$$R_{Y_0,\ldots,Y_n} = L_{Y_0,\ldots,Y_n} \cdot I_{Y_{n-1},Y_n} \equiv \prod_{i=0}^{n-1} L_{Y_i,Y_{i+1}} \cdot I_{Y_{n-1},Y_n} \qquad (1)$$

The cumulative dependency risk deems the overall risk exhibited by all the assets in the sub-chains of the n^{th}-order dependency. The cumulative dependency risk, denoted as $DR_{Y_0,Y_1,\ldots,Y_n}$, is defined as the overall risk produced by an n^{th}-order dependency:

$$DR_{Y_0,\ldots,Y_n} = \sum_{i=1}^{n} R_{Y_0,\ldots,Y_i} \equiv \sum_{i=1}^{n}\left(\prod_{j=1}^{i} L_{Y_{j-1},Y_j}\right) \cdot I_{Y_{i-1},Y_i} \qquad (2)$$

Equation (2) calculates the overall dependency risk as the sum of the dependency risks of the affected nodes in the chain due to a failure realised in the source node of the dependency chain. In order to compute the risk, a risk matrix that combines the likelihood and incoming impact values of each vertex in the chain is used. Interested readers are referred to Kotzanikolaou et al. (2013a) for additional details about dependency risk estimation.

But often the estimation of likelihood values is difficult or the required data for its estimation are not available. Therefore, while the identification of a dependency between two nodes is possible, the probability of a failure to propagate between the two nodes is either unknown or certain (likelihood = 1). In both cases, the following simplified version of equation (2), which follows the assumption that if a node fails, then the dependent nodes will also fail (likelihood = 1), is used:

$$DR_{Y_0,\ldots,Y_n} = \sum_{i=1}^{n} R_{Y_0,\ldots,Y_i} \equiv \sum_{i=1}^{n} I_{Y_{i-1},Y_i} \tag{3}$$

Then, the n^{th}-order dependency risk is calculated as the cumulative impacts on the affected nodes in the dependency chain.

3.2 Fuzzy logic combination of impact values

The equations (1) through (3) are based on the maximum expected impact of each dependency. Therefore, the multi-risk methodology described above is static in time. These equations do not take into account the factor of time and the values produced by them assume that:

1 each dependency chain will always produce its worst-case impact (and risk)

2 all the dependencies exhibit the same impact growth rate.

However, these assumptions are not met in reality. Neither do all nodes in a chain escalate to their maximum consequences nor do they experience the same impact growth rate over time. For this reason, the multi-risk methodology is extended to embody a dynamic, time-based analysis and to assess partial failure scenarios. Fuzzy set theory is used to model this behaviour.

Fuzzy set theory and FL, in contrast with classical set theory and classical logic, attempt to find approximations of ambiguous groupings in order to project objective evaluations of values requiring much effort and many resources to compute (CRAMM, 2010). FL variables may have a truth value that ranges in degree between 0 and 1. The goal is to use fuzzy approximations of impact evolution for various growth models in order to approximate the time evolution of a cascading failure, similar to a real failure. For instance, an incident might initially have a slow cascading effect on other dependent assets and, as time passes, a failure to restore operations might lead to catastrophic effects.

The main advantages of using FL is that it can work with no real-life training data since we know the domain we are modelling and its reaction/behaviour rules; e.g., we model chains of assets depending on business process needs and we know their impact/likelihood rules in case of failures through standard RA. The second advantage is FL's interpretability and simplicity, as it is used to "compute with words" and allows modelling near natural language rules. This is ideal in RA, where concepts of 'critical', 'high risk' or 'partial failure' are found in abundance. FL permits auditors to control these concepts through standard quantitative risk scales.

When new data or rules are added to the system, there is no need to re-train the system, mainly just adding new rules (besides rule conflict check).

3.3 A business process-based asset dependency analysis methodology

A business process is a step-by-step description of what users have to do to accomplish a specific task. Those steps utilise resources and assets from the IT system. Mapping these asset dependencies per business process allows us to calculate dependency chains and utilise them to assess the cascading and cumulative risk of potential threats on a business process.

In the context of business process management, risk has mainly been addressed as an overall factor to be considered during process-related evaluations (Rikhardsson et al., 2006). Our method proposes the use of decomposition of business processes into relevant assets and resources. Decomposition is used to identify business related assets inside the information system. This method can express business processes as a dependency tree of individual IT assets and services based on process characteristics. Traditional asset-based RA tools do not consider cross-functional asset dependencies that may support multiple business processes.

Essentially, the methodology realises five (5) steps while modelling asset dependencies per business process:

1 *Identify business processes* and relevant functional steps.

2 *Identify which assets* are utilised in each business process step.

3 *Create asset dependency chains* for prime asset of each process step. Each dependency gets an impact and likelihood value that depicts the overall risk of an asset failure for the depended asset. Values are calculated using traditional asset-based RA techniques.

4 *Combine asset chains* of all business process steps to create an asset dependency graph for the given business process.

5 *Calculate dependency risk chains* from the dependency graph and propose high-risk assets for targeted risk mitigation.

3.3.1 Example: train routing times business process modelled as asset dependencies

A railway operator has a specific business process for scheduling train routes. The organisation's business process steps could be the following:

a a user inputs train times to the railway scheduling (RS) software through an interface at his local workstation and requests scheduling information

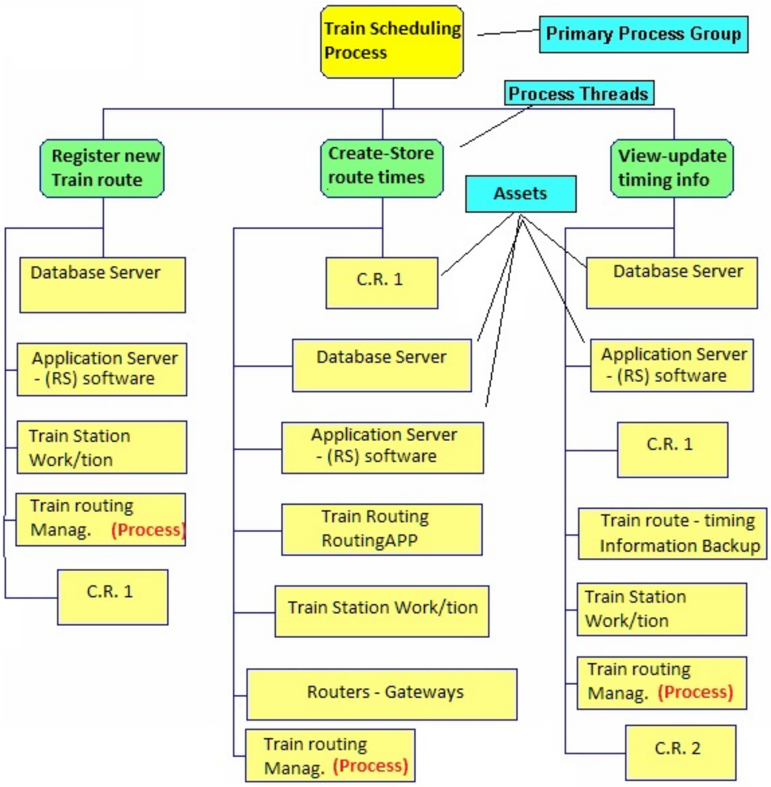
b RS software returns the train's overall route along with stop duration recommendations and speed recommendation from the railway routing system

c the user edits/updates information provided from the railway routing system d the RS software system sends the routing information to central server

e the RS software saves and shares the information with the employee and notifies the railway routing system; the system's software is the RoutingAPP

f if needed the user creates or provides routing timing adjustments though the RS software.

Figure 1 Example of a decomposition of a train scheduling process into asset uses (see online version for colours)



STEPS 1–2 Figure 1 shows an example of the above train scheduling process decomposition, able to depict business process dependencies on assets; CR1 and CR2 stand for computer room 1 and computer room 2.

STEP 3 Asset dependencies are modelled into chains based on each process step. The above decomposition gives us the following asset chains. Values in each dependency are indicative of the impact that a failure on the originating asset would have on the depended one. These values are calculated using traditional asset-based RA techniques.

An example of the asset dependency chains generated from the above business process threads are the following:

- CR1 → Application server → Railway scheduling (RS) softw. → Train station work/tion → Train routing. manag.
- CR1 → Database server → Railway scheduling (RS) softw. → Train station work/tion → Train routing manag.
- CR1 → Application server → Railway scheduling (RS) softw. → Routers → RoutingAPP
- CR1 → Database server → Railway scheduling (RS) softw. → Train station work/tion → CR2 → Backup
- CR1 → Application server → Railway scheduling (RS) softw. → Train station work/tion → CR2 → Backup

STEPS 4–5 Our methodology now combines the above asset dependency chains to form a dependency risk graph from which it will calculate all paths and detect the most serious risk chains based on the methodology and equations (2) and (3). It then proposes ways to lower the risk in these chains using potential countermeasures for risk mitigation.

4 Implementing the methodology

The tool utilises the Neo4jgraph database (Neo Technology, 2014) to model asset dependencies per business flow. Neo4J is widely considered highly adaptable, scalable and efficient (Jouili and Vansteenberghe, 2013; Shao et al., 2012) for these types of tools since it builds on the property graph model. Nodes have various labels that can serve as informational entities and are connected via directed relationships. Both nodes and relationships hold arbitrary properties (key-value pairs). Using the Neo4J technology, the proof-of-concept (P.O.C.) tool can represent complex graphs of even thousands of dependent assets through a weighted, directed graph. The proof-of-concept tool was developed using the Java language.

To demonstrate the applicability of our method, we utilised the tool to perform a full RA of a real-world infrastructure. All data are sanitised due to confidentiality agreements; therefore, labels and assets are renamed using generic terms and IDs. Still, impact, likelihood and time-related inputs assigned to each dependency are based on real RA results.

In this scenario, the infrastructure under test is comprised of two buildings and two computer rooms (CRs). The infrastructure was selected due to its size (relatively small and easy to model with about 200 workstations) and its well-documented cross-sectoral asset dependencies. Table 2 depicts the input data for interconnected assets that comprise the train routes scheduling business process of the infrastructure under test. The tool computes the complete set of asset dependency risk paths in a time frame for each dependency chain of order no greater than five using equation (3).

Besides printing all sorted dependency paths for each business process in excel files, the tool outputs a graphical representation of the examined dependency risk graph (an example can be shown in Figure 1). Darker colours in chains depict the maximum cumulative dependency risk path.

4.1 Real-world scenario – train routes scheduling business process

Let us consider the following business process used as a pilot for testing our RA methodology, which is based at a real-world infrastructure: At the aforementioned railway organisation, train routes, stop reschedules and timing information are all managed through the use of a relevant RS train routes and timing software. The program is responsible for storing data concerning train route times, query and provide timing information and inform employees about reschedules and route changes for delayed trains. The scheduling business process flow is comprised of the following steps:

1. a user inputs employee information to RS software through an interface at his local train workstation and requests routing information
2. RS returns the trains' and on time or delayed routing times from the central routing system.
3. the user creates/updates timing information and routing adjustments for his local station provided by the central routing system
4. if needed the user provides updated routing times and/or adjustments though RS
5. the RS system sends routing time notifications
6. RS saves and shares the information with the employee and stores it in secondary software, named RoutingAPP.

After analysing the organisation IT infrastructure, asset dependencies for this business process were modelled, as seen in Table 1.

Table 1 Asset dependencies for train routes scheduling business process

Initiating asset	*Dependency*	*Depended asset*
Server 1	Introduction of damaging or disruptive software (6)	Train routing RoutingAPP, railway scheduling (RS) software
Server 2	Introduction of damaging or disruptive software (6)	Railway scheduling (RS) software
Train routing RoutingAPP	Application software failure (4)	Train and routing data
Railway scheduling (RS) software	Application software failure (5)	Train and routing data
Routing building A	Introduction of damaging or disruptive software (6)	Server 1
Routing building A	Introduction of damaging or disruptive software (6)	Server 2
UPS power protection	Power failure (4)	Server 1
AC/DC generator	Power failure (4)	Server 2
Railway scheduling (RS) software	Application software failure (5)	Local train station data
Railway scheduling (RS) software	Application software failure (5)	Rail line data
Railway scheduling (RS) software	Introduction of damaging or disruptive software(6)	Train routing RoutingAPP

The asset dependency chains generated from the asset dependencies for the above business process are the following:

- CR1 → Routing building A → Server 1 → Railway scheduling (RS) software → Train and routing data
- CR1 → UPS power protection → Server 1 → Railway scheduling (RS) software → Train and routing data
- CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train and routing data
- CR1 → AC/DC generator → Server 2 → Railway scheduling (RS) software → Train and routing data
- CR1 → Routing building A → Server 1 → Railway scheduling (RS) software → Local Train Station Data
- CR1 → UPS power protection → Server 1 → Railway scheduling (RS) software → Local train station data
- CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Local train station data
- CR1 → AC/DC generator → Server 2 → Railway scheduling (RS) software → Local train station data
- CR1 → Switches building A → Server 1 → Railway scheduling (RS) software → Rail line data
- CR1 → UPS power protection → Server 1 → Railway scheduling (RS) software → Rail line data
- CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Rail line data
- CR1 → AC/DC generator → Server 2 → Railway scheduling (RS) software → Rail line data
- CR1 → Routing building A → Server 1 → Train routing RoutingAPP → Train and routing data
- CR1 → UPS power protection → Server 1 → Train routing RoutingAPP → Train and routing data
- CR1 → Routing building A → Server 1 → Railway scheduling (RS) software →Train routing RoutingAPP → Train and routing data
- CR1 → UPS power protection → Server 1 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data
- CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data
- CR1 → AC/DC generator → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data

4.1.1 Results and detection of highest risk business process chains

Table 2 depicts the input data for interconnected assets that comprise the train routes scheduling business process of the infrastructure under test.

Table 2 Input data for interconnected assets

Source Asset	*Depended Asset*	*Impact*	*Likelihood*	*Conn. type*	*Time – worst case scenario*	*Impact growth rate*
UPS power protection	Server 1	9	0.4	Physical	48 h	Slow
AC/DC generator	Server 2	9	0.4	Physical	24 h	Linear
Routing building A	Server 2	8	0.7	Physical	12 h	Fast
Routing building A	Server 1	8	0.7	Physical	12 h	Fast
Server 1	Train routing RoutingAPP	9	0.6	Informational	3 h	Fast
Server 2	Railway scheduling (RS) software	9	0.6	Informational	3 h	Fast
Server 1	Railway scheduling (RS) software	9	0.6	Informational	3 h	Fast
Train routing RoutingAPP	Train and routing data	9	0.4	Informational	24 h	Fast
Railway scheduling (RS) software	Rail line data	9	0.5	Informational	12 h	Fast
Railway scheduling (RS) software	Local train station data	9	0.5	Informational	12 h	Fast
Railway scheduling (RS) software	Train and routing data	9	0.5	Informational	12 h	Fast
Railway scheduling (RS) software	Train routing RoutingAPP	9	0.6	Informational	24 h	Fast

The tool combines the above asset dependency chains to form a dependency risk graph from which it will compute the complete set of asset dependency risk paths in a time frame for each dependency chain of order no greater than five using equation (3) and will detect the most serious risk chains, as seen in Table 3. Furthermore, the tool outputs a graphical representation of the examined dependency risk graph, as seen in Figure 2. In this case, this graph models asset dependencies that correspond to the flow of the train routes scheduling business process.

Table 3 and Figure 2 depict the output graph model of asset dependencies that correspond to the flow of the train routes scheduling business process. Ten asset nodes produced more than 40 dependency chains with orders ranging from two to five and with potential risk values between 6.1 and 11.36.

Table 3 Most serious dependency risk chains for the train routes scheduling business process

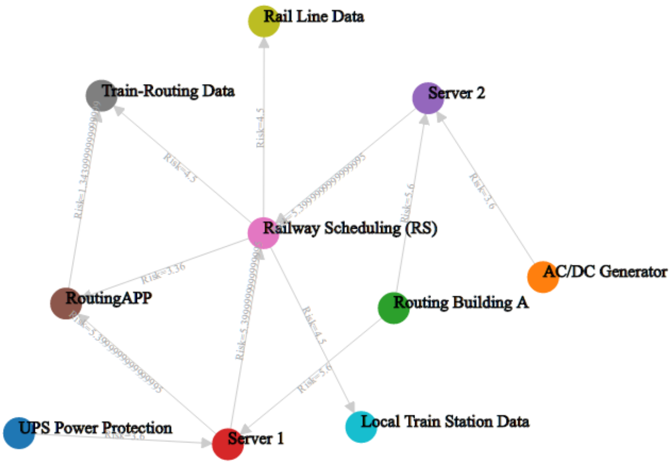
ID	*Time slot*	*Most serious risk chain*	*Cumulative risk chain for scheduling process*
#1	15 min	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Rail line data	6.1
#2	1 hour	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Rail line data	9.09
#3	3 hours	CR1 → Routing building A → Server 1 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	11.78
#4	12 hours	CR1 → Routing building A → Server 1 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#5	24 hours	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#6	48 hours	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#7	1 week	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#8	2 weeks	CR1 → Routing building A → Server 2 → Railway Scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#9	4 weeks	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56
#10	More	CR1 → Routing building A → Server 2 → Railway scheduling (RS) software → Train routing RoutingAPP → Train and routing data	12.56

The tool produced a list of all the dependency paths sorted according to the total cumulative risk of each one. Using this, potential RA auditors can identify all business processes with potential risk above a specified threshold value. The threshold parameter is subjective and defined by the decision maker and the particular characteristics of the infrastructure-under-assessment.

Someone can notice that, for the first time slots (i.e., if we manage to implement the business continuity plan and countermeasures sooner than 1 hour after a failure on the business process and relevant assets), the highest risk paths surpass a typical default risk value threshold of five (5); meaning that the train routes scheduling process is critically affected even 15 minutes after a malware infection due to potentially instant breach of confidentiality.

Thus, a cost-effective strategy to mitigate this business process risk would be to apply mitigation controls at node ‘routing controls at building A’ and at ‘server 2’ with a rapid response time; this would result in a substantial reduction in the overall cumulative risk for the train routes scheduling Business process. An interesting finding is that, for the first 24 h, server 1 is more critical than server 2 which still ends up having the highest overall risk impact on the total of dependency chains for the train routes scheduling business process.

Figure 2 Tool output: graphical representation of the examined dependency risk graph (see online version for colours)



A second result identified by the tool is that, although the asset dependency path

CR1 → Routing building A → Server 2 → Railway scheduling (RS) software
→ Train routing RoutingAPP → Train and routing data

exhibits the highest risk for almost all examined time slots, still, analysis revealed that paths #2 and #4 are the most critical about 1 h and 12 hours respectively after the cascading failure. This happens due to the fact that dependencies in those paths have rapid growths and are thus expected to have fastest convergence to maximum impact sooner than the aforementioned, most critical dependency path reaching the train and routing data.

4.1.2 Combined analysis of common-cause assets on the business process

Another finding the can be identified through the tool's output is that, for each examined node, all dependency paths that refer to the train routes scheduling Business process have been detected and calculated. In the examined case, this leads us to identify that (besides CR1 which is the obvious physical high-risk site), *RS* and *SERVER 1* are by far the most critical assets for the business process. The tool detected that:

1 the sum of distinct risk paths for each of these nodes is the highest (around 40)

2 these two assets exhibit the highest inbound and outbound connections (dependencies) for supporting the business process.

Note that the complete set of dependency chain risks is already an output of the tool. Thus, the evaluation of possible common-cause failures is based on 'ready-to use' risk chains.

5 Comparison with other approaches

Ouyang (2014) categorised CIP tools and methodologies using five main types of modelling and simulation approaches:

1 empirical-based

2 system dynamics-based

3 agent-based

4 network-based modelling approaches.

Recent surveys on modelling CI dependencies (Ouyang, 2014; Stergiopoulos et al., 2016b) showed that network-based modelling is used more than any other modelling technique [in Stergiopoulos et al. (2016b), 22 out of 34 presented CI modelling tools are utilising similar models to depict systems and asset dependencies].

The approach presented in this paper also draws from network-based methods in that it combines a method for discovering dependency risk paths with an automated modelling and analysis tool. It enables the dependencies of the connected infrastructures to be depicted as a graph and critical paths to be identified. As stated in D'Agostino et al. (2010), each infrastructure can be initially modelled as a mathematical object, a graph, consisting of different elements named nodes and arcs (or links) which are functional elements connecting the nodes. In D'Agostino et al. (2010), authors use seven basic interdependency indicators for topological and functional inter-dependency assessment.

Following Ouyang's classification (Ouyang, 2014), recent surveys on modelling CI dependencies (Ouyang, 2014; Stergiopoulos et al., 2016b) showed that network-based modelling is used more than any other modelling technique [in Stergiopoulos et al. (2016b), 22 out of 34 modern CI modelling tools presented are utilising graph-like models to depict systems and asset dependencies], mainly due to its ease to create abstract models of similar systems of systems present in CIs (Stergiopoulos et al., 2016a).

Such flow-based network approaches are described in the literature. They either model the flow of products or services between CIs in a uniform model (Svendsen and Wolthusen, 2007a, 2007b) or they combine various sector-based flow models (Santos and Haimes, 2004). Most modelling, simulation and analysis tools in the literature are sector-specific. For example, OpenMI (Talsma et al., 2012) supports federated modelling and simulation for the water sector. Other approaches allow for integrated or federated simulations that combine models from multiple sectors; examples include DIESIS (Ouyang, 2014; D'Agostino et al., 2010), EPIC (Santos and Haimes, 2004) and I2Sim (Rikhardsson et al., 2006).

Our proposed methodology also relies on empirical input from auditors and employees. Empirical methods for modelling CI dependencies have been criticised by researchers due to the lack of statistical data required to assess the likelihood of potential events. While probability data may be difficult to collect for many CIs, efforts have already been made to do so in specific C sectors. For example, Carreras et al. (2012) have conducted statistical studies of blackouts that enable the identification of critical power lines or groups of power lines for a given network model to identify lines likely to trigger or propagate cascading effects due to power line vulnerabilities.

Setola et al. (2009) approach also use FL to minimise the uncertainty and ambiguity associated with subjective information received from domain experts. On the other hand, our methodology combines FL with various time growth models. Each dependency may follow a different growth rate and FL is used to objectify the evolution of each dependency, taking into consideration the states of other nearby dependencies. This enables our methodology to output results for various time frames, not just economic dependencies. Other approaches (Santos and Haimes, 2004; Santos, 2006) use the input-output inoperability model to assess the dependencies between various sectors of an economy and to forecast the effects of a disruption in one sector on another sector. However, the approach presented in this paper is not a purely economic one.

Alpcan and Bambos (2009) developed a framework for analysing security risk dependencies in organisations and ranking the risks. The framework captures how risk 'diffuses' via complex interactions and reaches an equilibrium by introducing a risk-rank algorithm. To develop it, authors utilised bipartite graphs to represent the relationships between business units, people and security threats. Some of the differences with the proposed method are that the method in Alpcan and Bambos (2009) ignores intra-node risk-transfers and utilises a weighting system to provide risk vector calculations instead of FL.

Another important difference is that our methodology allows alterative graphs to be created to analyse dependencies that occur in abnormal operating conditions; in contrast, the inputs to the approaches described in Santos (2006) and Santos and Haimes (2004) only incorporate dependencies in normal economic operations. Additionally, our method can perform a time-based analysis, which offers different risk results according to the time frame studied and the rate at which the impact evolves in each CI.

Another economic-based approach is implemented in N-ABLE, a NISAC tool (Ehlen and Scholand, 2005). N-ABLE is a large-scale microeconomic simulation tool that models complex supply chains, spatial market dynamics and CI interdependencies between US businesses. N-ABLE is to model how US businesses adapt to and recover from disruptive events. CIDA, on the other hand, is not specifically engineered to model the economic impact at the microeconomic level.

The critical infrastructure protection/decision support system (CIP/DSS) (Bush et al., 2005; ISO, 2011), for example, is a complete RA methodology that can be applied to all sectors. Developed under the U.S. National Infrastructure Protection Plan (Department of Homeland Security, 2013), the methodology uses system dynamics with continuous time-step simulation. Like CIP/DSS, the CIPDSS-DM tool is designed to help analysts and policy makers evaluate and select optimal risk mitigation strategies. CIP/DSS and CIPDSS-DM are a robust combination. As a matter of fact, the ability of CIPDSS-DM to facilitate the selection of the most effective mitigation strategies is helpful in restricting the impact of failures and reducing economic losses. Previous experiments on our multi-risk dependency analysis methodology (Stergiopoulos et al., 2016a) reveal that our approach can efficiently compute the risks of all the dependency risk paths when reasonable limits are placed on the order of dependencies. However, the execution times for large-scale scenarios comprising hundreds of nodes may not be feasible for real-time analysis and response.

Other approaches state that one modelling technique is inefficient. IRRIIS (Klein, 2011) investigated a couple of different modelling approaches; authors believe that no single model is able to capture the different relevant aspects. Still, this type of multi modelling leads to very complex models that are rarely seen active in real-world tools (Stergiopoulos et al., 2016b).

6 Conclusions

In this paper, the concepts of asset dependency within business processes have been extended to create a new methodology for performing RA and risk mitigation in CIs. The findings derived from the dependency risk chains and the assessment results have been compared with respect to the importance of the assets and threats of the infrastructure-under-test. In our method, each asset dependency chain of the system has been assessed in terms of the business process it corresponds to. Thus, the representative dependency graph and the dependency risk measures have been computed. The risk chain measures have been shown capable of highlighting some IT safety strengths and weaknesses otherwise not detectable with typical RA methodologies. For example, identifying key assets that need to be more resilient allows for prioritisation of mitigation controls and also for minimising the cost of protection for the overall system. In this view, the time-based analysis of asset dependencies can constitute a valuable additional tool for the Risk assessors and managers to gain insights on IT resilience of infrastructure components and processes.

6.1 Limitations

A limitation of the methodology presented in this paper is its reliance on prior RAs of CIs. This is inherent to all the empirical risk approaches – empirical risk-based approaches analyse dependencies based on previous incidents (historical incident or disaster data) coupled with expert opinion to identify alternative measures that minimise the dependency risk (e.g., Franchina et al., 2011; Utne et al., 2011). It is unlikely for a single critical infrastructure owner or operator to have access to real data about other CIs. Thus, the methodology can only be applied at a higher layer. For example, sector coordinators or regulators may collect data about a specific sector such as energy or information and communications technology and disperse relevant sanitised data to infrastructures to aid them analyse their cross-sectoral dependencies. National critical infrastructure protection authorities may also be able to collect such information.

6.2 Future work

Future work will aim to combine the presented business-process methodology with a novel approach for calculating the Likelihood of occurrence of security incidents and threats to dynamically assess the evolution of cascading failures over time between assets involved in the interconnected business processes of multiple CIs. The likelihood metric will utilise historical data to chart complex mathematical distributions, able to provide a more objective Threat likelihood estimation than current, static ranked scales used in most modern RA methodologies.

As a case study, future work will utilise real-world cascading failure scenarios from CIs to test the proposed business-process RA with the novel Likelihood metric and compare it with current, RA results and relevant empirical knowledge from the CI's auditors to estimate potential advantages of the novel likelihood metric.

All experiments were performed using a computer with an Intel Core i7, 2.7 GHz processor with four cores and 16 GB RAM.

Acknowledgements

We would like to thank Despina Mentzelioti (AUEB) and Georgios Arvanitis (AUEB) for their useful insight and provided information. The contents of this article do not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

References

Agencenationale de la sécurité des systems d' information (ANSSI) (2010) *Ebios 2010 - Expression of Needs and Identification of Security Objectives* [online] http://www.ssi.gouv.fr/en/the-anssi/publications-109/methods-to-achieve-iss/ (accessed 18 September 2016).

Alpcan, T. and Bambos, N. (2009) 'Modeling dependencies in security risk management', *Proceedings of the 2009 Fourth International Conference on Risks and Security of Internet and Systems (CRiSIS 2009)*, pp.113–116, Toulouse, doi:10.1109/CRISIS.2009.5411969.

Amutio, M., Candau, J. and Manas, J.A. (2014) *MAGERIT – version 3, Methodology for Information Systems Risk Analysis and Management, Book I – The Method*, Ministerio de Administraciones Publicas.

Bush, B., Dauelsberg, L., LeClaire, R., Powell, D., Deland, S. and Samsa, M. (2005) 'Critical infrastructure protection decision support system (CIPS/DSS) project overview', *Proceedings of the Twenty Third International Conference of the System Dynamics Society*, pp.17–21.

Caralli, R., Stevens, J., Young, L. and Wilson, W. (2007) *Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process*, Software Engineering Institute, Massachusetts.

Carreras, B., Newman, D. and Dobson, I. (2012) 'Determining the vulnerabilities of the power transmission system', *Proceedings of the Forty-Fifth Hawaii International Conference on System Sciences*, pp.2044–2053.

Club de la Sécurité de l'Information Français (CLUSIF) (2010) *Mehari: Information Risk Analysis and Management Methodology* [online] http://www.clusif.asso.fr/en/production/mehari/index.asp (accessed 18 September 2016).

CORAS (2010) *A Guided Tour of the CORAS Method* [online] http://www.springer.com/cda/content/document/cda_downloaddocument/ 9783642123221-c3.pdf (accessed 18 September 2016).

CRAMM (2010) *CRAMM User Guide*, No. 5.2, UK Government, Security Service.

D'Agostino, G., Bologna, S., Fioriti, V., Casalicchio, E., Brasca, L., Ciapessoni, E., and Buschi, S. (2010) 'Methodologies for inter-dependency assessment', *2010 5th International Conference on Critical Infrastructure (CRIS)*, pp.1–7, Beijing, doi: 10.1109/CRIS.2010.5617578.

de Haes, S., van Grembergen W. and Debreceny, R. (2013) 'COBIT 5 and enterprise governance of information technology: building blocks and research opportunities', *Journal of Information Systems*, Vol. 27, No. 1, pp.307–324.

Dept. of Homeland Security (2013) *NIPP 2013: Partnering for Critical Infrastructure Security and Resilience*, USA.

EAR/PILAR – Risk Analysis and Management [online] http://www.ar-tools.com/en/tools/pilar/ (accessed October 2016).

Ehlen, M. and Scholand, A. (2005) 'Modeling interdependencies between power and economic sectors using the N-ABLE agent-based model', *Proceedings of the IEEE Power Engineering Society General Meeting*, pp.2842–2846.

ENISA (2006) *Risk Management: Implementation Principles and Inventories for Risk Management/Risk Assessment Methods and Tools*, Technical Department of ENISA Section Risk Management.

European Parliament, Council of the European Union (2016) *Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 Concerning Measures for a High Common Level of Security of Network and Information Systems across the Union* [online] http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L1148&amp;from=EN (accessed 18 September 2016).

Franchina, L., Carbonelli, M., Gratta, L., Crisci, M. and Perucchini, D. (2011) 'An impact-based approach for the analysis of cascading effects in critical infrastructures', *International Journal of Critical Infrastructures*, Vol. 7, No. 1, pp.73–90.

Giannopoulos, G., Filippini, R. and Schimmer, M. (2012) *Risk Assessment Methodologies for Critical Infrastructure Protection, Part I: A State of the Art*, JRC Technical Notes, Publications Office of the European Union, Luxembourg.

Gorod, A., Sauser, B. and Boardman, J. (2008) 'System-of-systems engineering management: a review of modern history and a path forward', *IEEE Systems Journal*, December, Vol. 2, No. 4, pp.484–499.

ISO (2011) *ISO/IEC 27005:2011, Information Technology – Security Techniques – Information Security Risk Management*.

ISO (2013) *ISO/IEC 27001:2013, Information Technology – Security Techniques – Information Security Management Systems – Requirements*.

Jamshidi, M. (Ed.) (2010) *System of Systems Engineering, Innovations for the 21st Century*, Wiley, 2010.

Jouili, S. and Vansteenberghe, V. (2013) 'An empirical comparison of graph databases', *Proc. of the International Conference on Social Computing*, pp.708–715.

Klein, R. (2011) 'The EU FP6 integrated project IRRIIS on dependent critical infrastructures', in Xenakis, C. and Wolthusen, S. (Eds.): *Critical Information Infrastructures Security. CRITIS 2010. Lecture Notes in Computer Science*, Vol. 6712, pp.26–42, Springer, Berlin, Heidelberg.

Kotzanikolaou, P., Theoharidou, M. and Gritzalis, D. (2013a) 'Assessing n-order dependencies between critical infrastructures', *International Journal of Critical Infrastructures*, Vol. 9, Nos. 1–2, pp.93–110.

Kotzanikolaou, P., Theoharidou, M. and Gritzalis, D. (2013b) 'Cascading effects of common-cause failures in critical infrastructures', in Butts, J. and Shenoi, S. (Eds.): *Critical Infrastructure Protection VII*, pp.171–182, Springer, Germany.

Kotzanikolaou, P., Theoharidou, M. and Gritzalis, D. (2013c) 'Interdependencies between critical infrastructures: analysing the risk of cascading effects', in Bologna, S., Hammerli, B., Gritzalis, D. and Wolthusen, S. (Eds.): *Critical Information Infrastructure Security*, pp.104–115, Springer-Verlag, Germany.

Neo Technology (2014) *Neo4j Graph Database*, California [online] http://www.neo4j.org (accessed 12 June 2017).

NIST (2011) *NIST SP. 800-39, Managing Information Security Risk: Organization, Mission, and Information System View*, Department of Commerce.

Ouyang, M. (2014) 'Review on modeling and simulation of interdependent critical infrastructure systems', *Reliability Engineering and System Safety*, January, Vol. 121, pp.43–60, ISSN 0951-8320 [online] https://doi.org/10.1016/j.ress.2013.06.040.

Platinum Squared (2014) *RiskSafe Assessment-Cloud Based Risk Assessment* [online] http://risksafe.co.uk/Usage/Operation (accessed 18 September 2016).

Rikhardsson, P., Best, P.J., Green, P. and Rosemann, M. (2006) 'Business process risk management and internal control: a proposed research agenda in the context of compliance and ERP systems', *Proceedings of Second Asia/Pacific Research Symposium on Accounting Information Systems*, 20 June, Melbourne.

Santos, J. (2006) 'Inoperability input-output modeling of disruptions to interdependent economic systems', *Systems Engineering*, Vol. 9, No. 1, pp.20–34.

Santos, J. and Haimes, Y. (2004) 'Modeling the demand reduction input-output (I-O) inoperability due to terrorism of interconnected infrastructures', *Risk Analysis*, Vol. 24, No. 6, pp.1437–1451.

Setola, R., De Porcellinis, S. and Sforna, M. (2009) 'Critical infrastructure dependency assessment using the input-output inoperability model', *International Journal of Critical Infrastructure Protection*, Vol. 2, No. 4, pp.170–178.

Shao, B., Wang, H. and Xiao, Y. (2012) 'Managing and mining large graphs: systems and implementations', *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp.589–592.

Simonsson, M., Johnson, P. and Wijkström, H. (2007) 'Mode-based IT governance maturity assessments with COBIT', *15th European Conference on Information Systems*, Switzerland.

Sousa-Poza, A., Kovacic, S. and Keating, C. (2008) 'System of systems engineering: an emerging multidiscipline', *International Journal of System of Systems Engineering*, Vol. 1, Nos. 1/2, pp.1–17.

Stergiopoulos, G., Kotzanikolaou, P., Theocharidou, M., Lykou, G. and Gritzalis, D. (2016a) 'Time-based critical infrastructure dependency analysis for large-scale and cross-sectoral failures', *International Journal of Critical Infrastructure Protection*, March, Vol. 12, pp.46–60, ISSN 1874-5482 [online] https://doi.org/10.1016/j.ijcip.2015.12.002.

Stergiopoulos, G., Vasilellis, E., Lykou, G., Kotzanikolaou, P. and Gritzalis, D. (2016b) 'Critical infrastructure protection tools: classification and comparison', *Proc. of the 10th International Conference on Critical Infrastructure Protection, (CIP-2016)*, March, USA.

Svendsen, N. and Wolthusen, S. (2007) 'Analysis and statistical properties of critical infrastructure interdependency multiflow models', *Proceedings of the IEEE SMC Information Assurance and Security Workshop*, pp.247–254.

Svendsen, N. and Wolthusen, S. (2007) 'Connectivity models of interdependencies in mixed-type critical infrastructure networks', *Information Security Technical Report*, Vol. 12, No. 1, pp.44–55.

Talsma, J., Becker, B., Gao, Q. and Ruijgh, E. (2012) 'Coupling of multiple channel flow models with OpenMI', *Proceedings of the Tenth International Conference on Hydroinformatics*.

Utne, I., Hokstad, P. and Vatn, J. (2011) 'A method for risk modeling of interdependencies in critical infrastructures', *Reliability Engineering and System Safety*, Vol. 96, No. 6, pp.671–678.