Probabilistic Mission Impact Assessment based on Widespread Local Events

Alexander Motzek, Ralf Möller and **Mona Lange**
Universität zu Lübeck
Institut für Informationssysteme
Ratzeburger Allee 160, 23562 Lübeck
GERMANY

email: {motzek,moeller,lange}@ifis.uni-luebeck.de

Samuel Dubus
Alcatel-Lucent, Bell Labs Research
Network Algorithm Routing and Security Program
Route de Villejust, 91625 Nozay
FRANCE

email: samuel.dubus@alcatel-lucent.com

Abstract *Assessing and understanding the impact of scattered and widespread events onto a mission is an ongoing problem. Current approaches employ score-based algorithms leading to spurious results. This paper provides a formal, mathematical model for mission impact assessment. Based on this model we reduce mission impact assessment of widespread local events to a well-understood mathematical problem. Following a probabilistic approach, we present a feasible solution to this problem and evaluate the solution experimentally. We put high care in only using actually available data and kinds of expertise.*

1 Introduction

Modeling dependencies of missions on various involved resources is a novel field of research, which pursues the goal of assessing the influences of local impacts on a higher goal, i.e. a mission. Early approaches use ad-hoc methods for impact assessment involving newly established algorithms

In this work, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise and bring them inline towards one well-defined mathematical model. Based on this mathematical model we find a well-understood mathematical problem: In a complex dependency network we find multiple widespread events, whose local effects must be assessed towards a global effect. Using a probabilistic approach, we can benefit from existing, well-defined and well-understood algorithms to solve this problem without returning spurious results.

We focus on actual feasibility of data acquisition and keep manual work to a minimum. We demonstrate and evaluate experimentally that our approach is of linear complexity with the size of application.

The rest of this paper is structured as follows: In Sec. 2 we develop a mathematical model for mission impact modeling based on views from different experts. Based on this model, we discuss mission impact assessment as a formalized problem, its theoretical complexity and give an experimental evaluation in Sec. 3. Being an emerging field of research, we give an overview of related work in Sec. 4. Sec. 5 gives a conclusion and outlook to future work.

2 Dependencies and Impacts

In the following, we take a view from different perspectives towards mission impact assessment. We consider three views from three experts from different expertise and bring them inline towards one well-defined mathematical model. Based on this mathematical model we find a well-understood mathematical problem that assesses a mission impact from widespread local events.

Every expert defines a different dependency model, where every modeled entity represents a random variable and a dependency between two entities is represented by a local conditional probability.

Remark 1. *The here presented approach was developed in a business focused use case. Instead of referring to* missions, *we refer to* business processes *in a company and we use both terms interchangeably. Every occurrence of a "business" resource should be adaptable to a "mission" resource.* ▲

2.1 Mission Dependency Model (Business View)

In the field of business intelligence, a complete company or organization, i.e. a good we aim to protect, is modeled as a conglomeration of *business processes*. Commonly, business processes are modeled using the business process modeling notation (BPMN) and a business process is modeled as a (dependent) collection of tasks. This modeling approach is well accepted and can be found, e.g. in [2, 1, 8]. Fig. 1 shows a sketch of a BPMN model used throughout this paper.

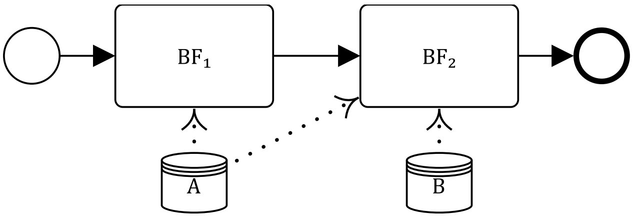

Figure 1: Example BPMN 2.0 model sketch for the BP_1 business process shown in the dependency model of Fig. 2.

Designing BPMN models is handled manually by an expert from a company or by an external business consultant having a precise expertise in the understanding of business analysis. The business analysis is performed on a pure business perspective and stops at a "device" level, e.g. it identifies a web-service, but does not describe the dependencies of the webservice on a database or a data center. This is a reasonable approach, as the latter perspective comes from a very different expertise and would require very broad-range experts. Further, an identification of a "web-service" as a business relevant object is precise in the terms of a business perspective, as, if the web-service is not running, the business process might not be accomplishable. From an "IT" perspective, the web-service might be irrelevant, as the crucial point of failure lies in the availability of data from a database. The latter dependencies are covered in the upcoming subsection.

We extend [7] and we model mission dependencies as shown in Fig. 2. We model a *company* as being dependent on its *business processes*. A business process is again dependent on one or more *business functions*. *Business devices* provide business function. Business devices are part of the network perspective and—from a network perspective—might be irrelevant, but were identified to be business critical. Fig. 2 shows a dependency graph of business relevant objects, based on the preceding presented BPMN model.

We model every dependency as local conditional probabilities. Every conditional probability describes the probability of failure if a dependency fails. E.g. the probability of the business-function BF_1 (see Fig. 2 and Fig. 1), e.g. "provide access to customer data", failing, given the required business-device A, e.g. "customer-data database", is 0.9.

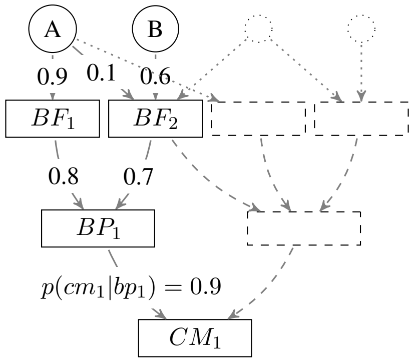

Figure 2: Mission Dependency Model. Values along edges denote individual conditional probability fragments. For our example only the solid entities are used. Consequences and further attributes are omitted in this figure.

Definition 1 (Probabilistic Preliminaries)**.** *We represent every node inside our dependency models by a random variable, denoted as capital X, where every random variable is assignable to one of its possible values $x \in \mathrm{dom}(X)$. Let $P(X = x)$ denote the probability of random variable X having x as a value. For our case we consider $\mathrm{dom}(X) = \{true, false\}$ and we write x for the event $X = true$ and $\neg x$ for $X = false$.* ▲

The event x represents the case that node X is operationally *impacted* and $\neg x$ that is is working at its fully operational capacity, i.e. no impact is present.

Definition 2 (From Dependencies to Distributions)**.** *We render every dependency of random variable Y on X as an* individual *conditional probability $p(x|y)$ and $p(x|\neg y)$. Such individual conditional probability are* fragments *of a complete conditional probability distribution and are therefore denoted in lowercase. To acquire the local conditional probability distribution $P(X|\vec{Y})$ of node X from* all *its individual dependencies $p(X|Y)$ of all dependent nodes $Y \in \vec{Y}$, we employ a non-leaky noisy-or combination function [10, 6]. Non-leakiness implies $p(x|\neg y) = 0$ for every dependency and therefore $P(x|\neg\vec{y}) = 0$.* ▲

With Def. 2, we obtain a Bayesian network from our mission dependency model, for which we can specify a joint probability distribution over all entities in the dependency model plainly as the product of all local conditional probability distributions.

The example given below shows how we can use a probabilistic dependency model as a Bayesian network and how an impact can be assessed.

Example 1. *Following the rather simple mission dependency model depicted in Fig. 2 (excluding dashed entities), we obtain the joint probability distribution $P(CM_1, BP_1, BF_1, BF_2, A, B)$ as*

$$= P(CM_1|BP_1) \cdot P(BP_1|BF_1, BF_2) \cdot P(BF_1|A) \cdot P(BF_2|A, B) \cdot P(A) \cdot P(B) \;, \tag{1}$$

where $P(BP_1|BF_1, BF_2)$ and $P(BF_2|A, B)$ are obtained through the noisy-or assumption from $p(bp_1|bf_1)$, $p(bp_1|bf_2)$ and $p(bf_2|a)$, $p(bf_2|b)$ respectively.

We can then marginalize the conditional probability of a mission impact on CM_1 from, say, an observed impact on $A = a$ and none on $B = \neg b$ as

$$P(cm_1|a) = \alpha \cdot \sum_{BP_1} \sum_{BF_1} \sum_{BF_2} P(cm_1, BP_1, BF_1, BF_2, a, \neg b) \;, \tag{2}$$

with a normalizing factor α, s.t. $\sum_{CM_1} P(CM_1|a) = 1$. ♦

To detail an effect of an impact, we define a set of *consequences* for every business process to which a possible failure of the business process might lead. Again, we model a consequence as an individual conditional probability stating the probability that a consequence happens, given an impact on the business process. Likewise, we can then calculate the probability that a BP's consequence happens (con_{BP}), given all observed local impacts, say a, plainly as $P(con_{BP}|a) = P(con_{BP}|bp) \cdot P(bp|a)$.

Still, only considering a business view does not cover transitively (or passively) involved resources. To cover distant and widespread local events, which are not directly obvious, we introduce a network dependency model in the upcoming subsection.

2.2 Network Dependency Model (IT View)

As mentioned afore, an identified critical device might be threatened *transitively* by further devices inside the network. In a network modeled by an IT expert we cover dependencies between individual network nodes, which can be, e.g., individual ICT servers, ICS devices, software components or other operationally needed resources. We follow the same "Bayesian" approach as before, i.e. every dependency between two devices represents a local conditional probability of failure, if the dependence fails, as shown in Fig. 3.

However, in contrary to the mission dependency model, assessing network dependencies might not be manageable by hand. Complex network architectures render a manual dependency analysis infeasible and error prone. Further, new dynamically adjusting network architectures make it even unknown to an expert to identify exact network dependencies. However, it is possible to validate a presented network dependency model for plausibility. We therefore employ heuristics based on exchanged information amounts, e.g. traffic analyses, to identify possible network dependencies. As long as a network device only consumes relevant information for its purpose, every data transfer inside the network must motivate some dependency. Moreover, collecting traffic information about a network is a reasonable and feasible effort. Further, under the assumption of *per node* equally distributed entropy and encoding of consumed information, a dependency, i.e. a conditional probability, must be a function of consumed information bits. We, therefore, reduce an infeasible effort of identifying all dependencies by hand onto finding a heuristic, or rather, checking a generated dependency model.

Example 2. *In our use case, we have information about exchanged information at a logical ICT device level covering virtual machines as individual devices. More granular data, e.g. on software layers, was not acquirable. Fortunately, we can assume in our use case that every device drives one purpose. Multiple software applications running on one device will most likely be dependent on each other, and a failure of one software component will very likely lead to a failure of other software components. We can say that dependencies at device level are coarse enough.*

For example, a workstation X consuming different query results from multiple databases will distribute gained and processed information from such queries to other devices. The percentage of received traffic $T_{Y_i,X}$ from every database Y_i towards the total received traffic can give us a good guideline for the conditional dependency between them as $p(x|y_i) = \frac{T_{Y_i,X}}{\sum_i T_{Y_i,X}}$. However, if the workstation further consumes irrelevant 5TB of cat pictures from a local file server, the heuristic will fail, because the workstation also consumed many irrelevant information. Depending on a network or company characteristics other heuristics might be appropriate, e.g. derivation from a mean received amount of data or a mapping onto a σ distribution. ♦

2.3 Local Impacts (Security View)

A third view involves a security expert able to assess local consequences of events. In the style of reliability analyses using Bayesian approaches we model external shock events inside a network. Every node X might

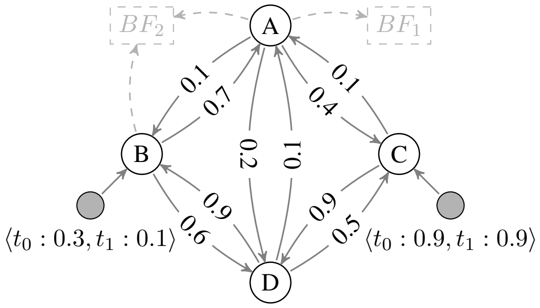
BF_2
A
BF_1
0.1
0.7
0.1
0.4
0.2
0.1
B
C
0.9
0.6
0.9
0.5
D
$\langle t_0 : 0.3, t_1 : 0.1\rangle$
$\langle t_0 : 0.9, t_1 : 0.9\rangle$

Figure 3: Network Dependency Model. Dependencies between B, C would also be possible. Conditional probability fragments are marked along the edges. Grey nodes represent external shock events leading to local impacts. The time-varying conditional probability of local impact given an instantiated external shock event is given below an event's node. Connections to the mission dependency model are sketched in dashed grey.

be affected by one or more external shock events $\vec{SE}$, which are prior random variables. An external shock event $SE \in \vec{SE}$ might be present (se) or not be present ($\neg se$), for which a prior random distribution $P(SE)$ is defined. In the case that an external shock event is present (se), there exists a probability of it affecting a node X, expressed as a local conditional probability fragment $p(x|se)$. If an external shock event exists and it is not inhibited, we speak of a *local impact* on x. In the case that the external shock event is not present, i.e. $\neg se$, it does not affect random variable x and we write $p(x|\neg se) = 0$. Every individual conditional probability fragment from an external shock event is treated in the same noisy-or manner as a dependency towards another node, and thus, multiple shock events can affect one node. We consider fully obseravble external shock events. Extensions to partially observable shock events are straightforward.

Classically, a local impact can also be seen as an observation of an impacted node, i.e. x. However, in a Bayesian approach this would imply that this impact origins from inside the modeled network and would "blame" other nodes for it. By introducing external shock events we gain the ability to model "soft evidence" of local impacts, i.e. we are not sure whether an external shock event might actually lead to a local-impact and affect a node's operational capability.

Definition 3 (Temporal Aspects)**.** *We define a temporal aspect of an external shock event. We employ the idea of abstract timeslices in which the effect of an external shock event changes. Every abstract time slice then represents a duplicate of the network- and mission dependencies with a different set of local conditional probabilities of local impacts. We denote time-varying probabilities in a sequence notation as* $\langle t_0 : p_0, \ldots, t_T : p_T\rangle$*, where we have* $T + 1$ *abstract timeslices. In every abstract timeslice* i*, varying local impacts take their respective probability* p_i *defined for its time slice* t_i*.* ▲

Every local impact represents a potential threat and can be, for example, a consequence of a present vulnerability, a countermeasure, an attack, or originate from hardware failure. It lies in the expertise of a security operator to assess a potential *local* impact of those threats. Note that he does not need to have neither any expertise in network dependencies nor an understanding of missions to do so. The following Ex. 3 shows an example on how external shock events can lead to local impacts in a security context on selecting an adequate response plan to an (ongoing) attack.

Example 3 (Response Plan Side Effects)**.** *We employ mission impact assessment to achieve a* qualitative *assessment of potential negative side effects of a proposed response plan to an ongoing or potential attack. We see a response plan as a collection of individual actions affecting a network. E.g., a shutdown of a server might easily reduce the surface of a potential attack. Still, if a critical device is highly dependent on that server, it might impact a mission even heavier than a potential attack. We consider three mitigation-action types and transform them to external shock events, possibly leading to local impacts.*

The first mitigation action, i.e. an external shock event, is a "shutdown". Obviously, if a node is shut down (se: the external shock event is present) we can easily say that the probability of local impact, given the shutdown of node X, is 1, i.e. $p(x|se) = 1$.

Second, employing a patch on a node X might produce collateral damage as well. During installation of the patch, there might be a (low) probability of immediate conflict. In a mean time, a patch might enforce a reboot of a device. This leads to a temporal shutdown and might lead to hardware failure. Finally, after a successful reboot, a replacement of hardware, and/or a restore of a previous backup, the device will fully resume its operational capability. Using temporal aspects, we can model a patching operation in three abstract time slices and define the local impact probabilities of this external shock event to be $p(x|se) = \langle t_0 : 0.1, t_1 : 1.0, t_2 : 0.0\rangle$.

Our third considered mitigation action is the restriction of a connection from node X to node Y, i.e. a new firewall rule. From a technical perspective this operation forbids a transfer of data that might have been crucial for the operational capability of a node Y. Therefore, a firewall rule leads to an operational impact on Y. We must assess this impact locally. This is a special case requiring Pearl's [9] do-calculus. As a connection between two devices resembles a dependency, we must further actually remove this dependency. Otherwise, we would infer further impacts over a dependency that was prohibited and already assessed locally. To do so, we simply "bend" the forbidden dependency to an external shock event se, s.t. the local conditional failure probability $p(y|x)$ becomes a local impact probability $p(y|se)$. Another approach, decidable by a security operator, would be to accumulate dropped connections and add an unified local impact for them. ♦

3 Mathematical Mission Impact Assessment

Informally speaking, we have a mission dependency network and a device dependency network. In the device dependency network, some nodes are threatened by external shock events. As nodes are dependent, a threatened node might again threaten another node. We say, a node is threatened by an external shock event *transitively*. This leads to a "spreading" of external shock events. In the end, there exists a probability that even a business process or the complete modeled company (mission) is threatened transitively by various external shock events. To recall, to be threatened by an external shock event (might) lead to an impact; and it is a well-defined problem of calculating this "might"-probability of being impacted due to an external shock event, which is what we call the mission impact assessment.

Definition 4 (Mission Impact Assessment)**.** *The probability of a mission node MN being impacted, is defined as the conditional probability of MN being impacted mn given all observed external shock events $se \in \vec{se}$, i.e. $P(mn|\vec{se})$, where the effects of local impacts due to $\vec{se}$ are mapped globally based on mission-dependency and network-dependency graphs.* ▲

Given Def. 4 it is the task of mission impact assessment to calculate the probability $P(mn|\vec{se})$. To calculate this, multiple established approaches are available. From a probabilistic graph view, we need a sound definition of an overall joint probability distribution as demonstrated in Ex. 1. This is well defined for the mission dependency graph, because it is a directed acyclic graph and is a Bayesian network. However, in the network dependency graph we cannot assume an acyclicity constraint and a joint probability distribution is not defined for cyclic graphs. We could therefore, transform the network dependency model into a dynamic Bayesian network and perform a filtering operation on it. However, this introduces a high modeling overhead. Further, we could see the network dependency graph as a Markov random network, which, however, due to a needed global normalization factor, destroys an intended local view on probabilities. Due to the employed noisy-or assumption, we can view the graph and problem as a probabilistic logic program determining the

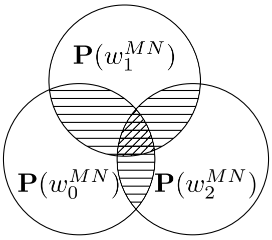

Figure 4: Illustration of $\mathbf{P}(w_i^{MN})$ viewed as sets. Overlapping parts (filled with patterns) are commonly shared probabilities along one path (see Def. 5) are not allowed to be counted twice (or even multiple times) when calculating $\bigcup_i \mathbf{P}(w_i^{MN})$.

probability of connectivity between a mission node and external shock events. This is a probabilistic path search.

This means to calculate the conditional probability of $P(mn|\vec{se})$, every path w_i^{MN} from an external shock event $se \in \vec{se}$ to the mission node MN is a chain of probabilities and is sufficient to induce $\{MN = true\} = mn$. Every path exists with a probability $P(w_i^{MN})$, where $P(w_i^{MN})$ is the product of all probabilities in this path. Let $\mathbf{P}(w_i^{MN})$ denote the probability viewed as a set. $P(mn|\vec{se})$ is then the probability that at least one path exists. I.e.

$$P(mn|\vec{se}) = P(\bigvee_i w_i^{MN}) = \bigcup_i \mathbf{P}(w_i^{MN}) \, , \tag{3}$$

where not all $\mathbf{P}(w_i^{MN})$ are disjoint (see Fig. 4) and it is worth noting that all paths might share common "edges". As every edge represents a probability, plain summation would double count these probabilities and lead to spurious results. This is exactly the issue from which many fudge-factor based "propagation" algorithms in ad-hoc solutions suffer.

An exact calculation of $\bigcup_i \mathbf{P}(w_i^{MN})$ is possible by the inclusion and exclusion principle and the Sylvester-Poincaré equality. Still, calculation is exponentially hard due to the subtraction of all overlapping sets and is therefore not practical. We therefore approximate the result by the use of a Monte-Carlo simulation.

3.1 Monte-Carlo Approximation

In order to approximate $P(mn|\vec{se})$ we employ a two-step simulation. As discussed in this section we calculate the conditional probability through a probabilistic path search. We first acquire all paths leading to external shock events, for every mission node for which we would like to perform mission impact assessment. Often, we would like to perform this for every node in the mission dependency model. Finding paths for a node in the mission dependency graph is trivial, given found paths from business devices to external shock events. We therefore, as step one, acquire (all) paths leading to evidence for all business devices, which is a classic graph search problem. Under the assumption that the number of business devices and external shock events is comparably small to all nodes in the network graph, a depth-limited search is a reasonable approach for finding paths leading to external shock events.

Definition 5 (Probabilistic Paths)**.** *For every business device $BD_i \in \vec{BD}$ let $\vec{w}^{BD_i}$ denote the set of all paths leading to an external shock event and let $w_j^{BD_i}$ denote the jth path. Let $\vec{w}$ denote the super-set of all found paths. Every path $w_j^{BD_i}$ is a set of individual conditional probability fragments $p(x|y)$, representing an edge, i.e. a dependency, from y to x. The product of all probability fragments $p(x|y) \in w_j^{BD_i}$ is the exist-probability of a path $P(w_j^{BD_i})$. Every path $w_k^{BD_i}$ for which holds $\exists j : w_j^{BD_i} \subseteq w_k^{BD_i}$ is irrelevant for calculation and $\vec{w}$ is a finite set. Informally this means, during path search along one path an already visited node must not be visited again and we cannot get stuck in infinite loops.* ▲

After acquiring all paths $\vec{w}$ leading to all business devices, subsequent paths leading to business functions, processes and the company are trivially acquired by following the paths leading to all children.

Step two is a Monte-Carlo simulation to approximate $P(\bigvee \vec{w}^{BD_i})$ for every business device $BD_i \in \vec{BD}$. We draw a sample from $\vec{w}$ and from all dependencies in the mission dependency model. We check for every BD_i the satisfaction of $\bigvee \vec{w}^{BD_i}$ and mark the satisfaction result on BD_i. Subsequently we check for satisfaction of any children, i.e. dependencies, of every node in the mission dependency model. Every satisfaction for a mission node MN found in the mission dependency model is marked as a hit in hit_{MN}. After n_{roll} times, the desired conditional probability of MN being impacted (mn), i.e. the *mission impact*, given all external shock events $se \in \vec{se}$ is approximated by $P(mn|\vec{se}) = \frac{hit_{MN}}{n_{roll}}$.

Remark 2 (Path Check)**.** *Checking all paths during one Monte-Carlo round is highly optimizable. $\vec{w}^{BD_i}$ can be sorted descending by $P(w_j^{BD_i})$, s.t. most likely existing paths are checked first and subsequent checks can be skipped once a path is found. Further, a path $w_j^{BD_i}$ can be sorted ascending by its individual local conditional probability fragments, s.t. most unlikely random variables are checked first and further checks inside one path can be skipped. Notwithstanding, the complete process is highly parallelizable.* ▲

Remark 3 (Temporal Aspects Implementation)**.** *We introduced that evidence, i.e. an external shock event, can have different conditional local probabilities depending on an abstract time slice. This means we have a varying probability at the end of one path $w_j^{BD_i}$. Naively, we could perform a Monte-Carlo simulation for every abstract time slice. However, this would redundantly simulate all non-varying probabilities. We therefore partition $w_j^{BD_i}$ in a non-varying set of conditional probabilities, i.e. a network path leading to an impacted node, and a set of varying conditional probabilities, i.e. a set of local impacts.* ▲

The following example gives a short demonstration of mission impact assessment using the defined mathematical model using an approximate Monte-Carlo method.

Example 4. *Let us consider Fig. 3, where an identified mission critical device A (compare Fig. 2) is threatened (transitively) by local impacts on nodes B and C. Let us call the local impacts SE_B and SE_C. Let us exclude the dependency of BF_2 on B and temporal aspects for brevity. Through depth-first search we find the paths $\vec{w}^A$ as*

$$\begin{aligned} w_0^A &= \{p(a|b), p(b|se_B)\} \\ w_1^A &= \{p(a|b), p(b|d), p(d|c)p(c|se_C)\} \\ w_2^A &= \{p(a|c), p(c|se_C)\} \\ w_3^A &= \{p(a|c), p(c|d), p(d|b), p(b|se_B)\} \end{aligned} \tag{4}$$

Additional paths, e.g. $w_\emptyset^A = \{p(a|b), p(b|c), p(c|b), p(b|se_B)\}$, are redundant, as, here, w_0^A is always (already) satisfied, if $w_\emptyset^A$ is satisfied. After finding these paths, finding paths to higher nodes in a mission dependency model, say, to BF_1, is trivial, by simply appending $p(bf_1|a)$ to every path of A. Subsequently, the same holds for BP_1 and CM_1.

For the simulation, at first every used random variable is sampled. Let $\vec{RV}$ be the vector of all random variables included in all paths. I.e. $\vec{RV} = \langle p(a|b), p(b|se_B), p(b|d), p(d|c), p(c|se_C), p(a|c), p(c|d), p(d|b), p(bf_1|a), p(bf_2|a), p(bp_1|bf_1), p(bp_2|bf_2), p(cm_1|bp_1)\rangle$. Let $\vec{rv}$ denote a sample of $\vec{RV}$, say, $\vec{rv} = \langle +, +, +, +, +, -, -, -, +, +, +, +, +\rangle$, where $+$ represents a $true$ sample, and $-$ a $false$ sample.

Subsequently, for every identified critical device, i.e. A, we check if at least one of its path is satisfied, i.e. if $\bigvee \vec{w}^A$ is satisfied. We obtain that w_0^A is satisfied and this satisfies A. The circumstance that w_1^A is also satisfied, but w_2^A and w_3^A are not satisfied is irrelevant and further checks can be skipped. Subsequently, we can check the remaining mission dependency graph for further satisfactions in this sampling round. As A and

$p(bf_1|a)$ are satisfied, BF_1 is satisfied as well. The same holds for BF_2. Likewise, BP_1 is satisfied as well as CM_1. Every satisfaction is marked as a successful Monte-Carlo round and increments a mission node's MN hit counter hit_{MN}.

This procedure is repeated n_{roll} times, i.e. $\vec{rv}$ is sampled and $\vec{w}$ is checked. Finally, every operational impact assessment of a mission node MN, represented by the conditional probability $P(mn|se_C, se_B)$, is approximated by $P(mn|se_C, se_B) = \frac{hit_{MN}}{n_{roll}}$. ♦

3.2 Complexity and Experimental Evaluation

We implemented the proposed approach as a flexible framework allowing user defined definitions of local impacts, user defined heuristics for dependency approaches and user defined performance characteristics as defined below. As central theme, we focused on actual feasibility of our proposal and we demonstrate that our approach scales well, i.e. linear, with a graph's complexity. In the following, we give a short expected summary of the complexity of our approach and evaluate it experimentally.

Evaluation and demonstration of the computational complexity of our presented approach is difficult, as it depends on the graph structure of the network and the processed response plan. We therefore use random graphs containing n_N nodes and $n_E = n_N^2 \cdot 0.1$ edges while assuring that every node is at least bidirected. By doing so we obtain a fully connected graph with, approximately, a 10% chance of two nodes being directly connected. The processed response plan consists of n_{MA} randomly placed mitigation actions (external shock events). For evaluation we place rather many $n_{MA} = n_N \cdot 0.1$ mitigation actions, i.e., 10% of all nodes are possibly impacted. We measure the time t_{ps} required for finding all n_P paths up to depth d_{max}, and t_{sim} required for simulating all found paths n_{roll} times. Every measurement is repeated in 50 different random graphs.

Complexity is differentiated between both steps of the impact assessment. Given a constant maximum search depth d_{max}, depth-limited search (DLS) scales linearly with the number of edges n_E, as also experimentally evaluated in Fig. 5. Further, DLS scales slightly with the number of placed local impacts n_{MA} (compare Fig. 10), as a pre-computation of shortest distances to local impacts per node can eliminate dead ends early. We write, path search is a function as $t_{ps} = f(n_E, d_{max}, n_{MA})$.

Remark 4 (DLS)**.** *Obviously, DLS scales exponentially with a specified maximum depth d_{max}. In general and for our example, the maximum depth should be chosen in the range of the average path length inside a given graph, s.t. almost every node is considered at least once. In order to better scale a maximum depth it is reasonable to allow a rational d_{max}, where a depth $d_{dec} < 1$ resorts to the best d_{dec}%, i.e. most dependent, children.* ▲

Monte-Carlo simulation, i.e. checking of paths, scales linearly with the number of found paths n_P (compare Fig. 6 and 7) and the number of Monte-Carlo rounds n_{rolls} (compare Fig. 8), i.e. $t_{sim} = f(n_P, n_{roll})$. Naturally, the number of proofs n_P scales with the number of local impacts n_{MA}, of edges n_E and the maximum path length d_{max} (compare Fig. 9), i.e. $n_P = f(n_{MA}, n_E, d_{max})$.

In summary, we conclude that experimental results match theoretically expected complexities.

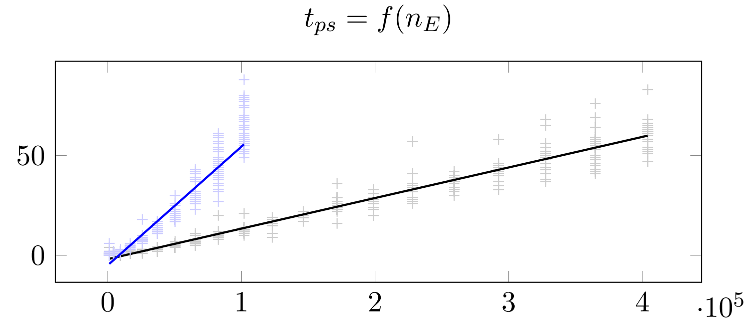

Figure 5: Pathsearch is linear with the number of edges in the graph. Black: $d_{max} = 3$, Blue: $d_{max} = 4$. n_N is linearly increased, meaning a quadratic increase of edges. $t_{ps} = f(n_E, d_{max}, n_{MA})$, n_{MA} only very slightly. t_{ps} in ms.

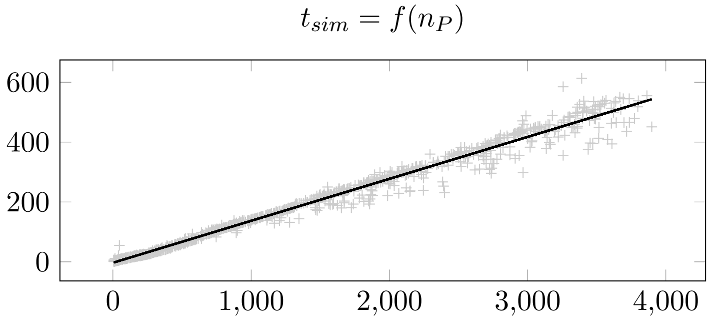

Figure 6: Simulation time is linear with the found paths. $d_{max} = 3$. n_N is linearly increased, meaning a linear increase of mitigation actions and a quadratic increase of edges, both increasing the amount of findable paths. t_{sim} in ms.

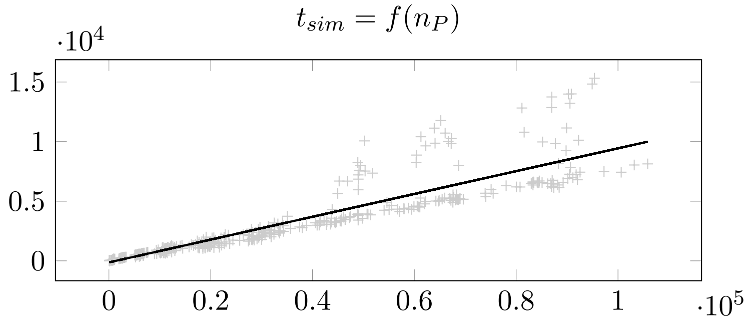

Figure 7: Same simulation as Fig. 6, but for $d_{max} = 4$. Linear time complexity is also achieved for very large proof sets.

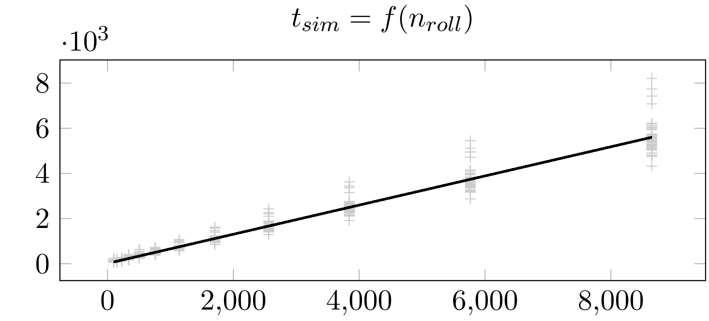

Figure 8: Simulation time directly correlates with the number of rolls. $n_N = 500$, $n_{MA} = 50$, $d_{max} = 4$. **Around** 12000 **proofs are found each time.** n_{roll} **is increased exponentially and every measurement is repeated for 50 different random graphs.** t_{sim} **in ms.**

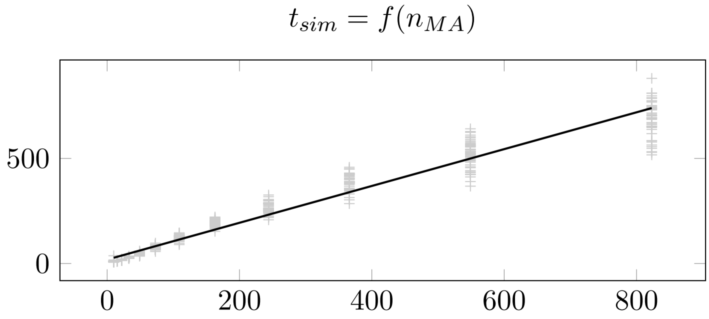

Figure 9: Simulation time directly correlates with the number of mitigation actions. Constant $n_N = 1000$, $d_{max} = 3$, $n_{roll} = 1000$. n_{MA} **is increased exponentially and repeated in 50 different random graphs. The more MAs, the more paths, the longer it takes.** t_{sim} **in ms.**

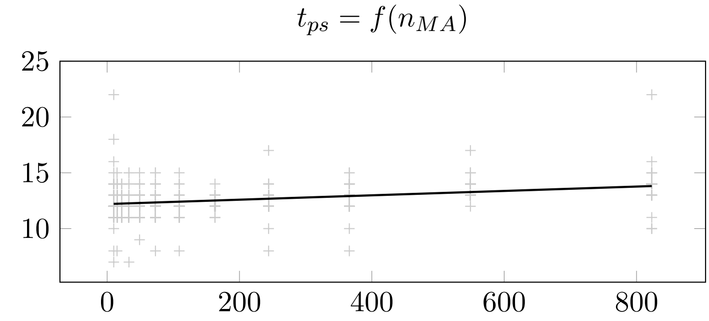

Figure 10: Proofsearch time is negligible dependent of the number of mitigation actions. t_{ps} **correlates with the number of edges in the graph. Measurements collected during Fig. 9.** t_{ps} **in ms.**

4 Related Work

Mission modeling and mission impact assessment is an emerging field of research; and, naturally in new, viral research areas, employ ad-hoc solutions using algorithms involving fudge factors. While delivering early results and acclaimed solutions for mission impact assessment, a formal definition of an underlying problem is yet missing. Employed fudge factors in newly established algorithms lead to untraceable and spurious results demanding data driven validations. Unfortunately, large, standardized datasets for validation are yet missing for mission impact assessment and in the following presented work. In the following, we point out valuable approaches and ideas.

Barreto et al. [2] introduce a well-understood modeling technique and use BPMN models to acquire knowledge. An impact assessment is based on various indexes and numerical scores, such as exploit index, impact factor, infrastructure capacity index, and graph distances. Albanese et al. present in [1] a well-modeled formalism for complex inter-dependencies of missions as a set of tasks. Using numerical scores and tolerances in a holistic approach Albanese et al. focus on cost minimization. Buckshaw et al. [3] propose a quantitative risk management by involving various experts and present a score-based assessment based on individual values and a standardization using a weighted sum.

Jacobson [7] presents a well understood conceptual framework using interdependencies based on operational capacity. In this dependency model, impacts are propagated and reduce the operational capacity. [7] uses self-defined metrics for propagating impacts through Boolean gates.

Further works focused solely on modeling. E.g., Goodall et al. [5] focus on modeling and available data integration using ontologies but do not address an impact assessment. Another ontology-based approach is presented by D'Amico et al. in [4] and identifies multiple experts while noting that, e.g., system administrators are not capable of understanding an organization's missions.

Notwithstanding, we were inspired by several aforementioned modeling ideas, such as using the BPMN standard and we considered different views from various experts. To the best of our knowledge, we contribute a novel, formalized, mathematical mission impact assessment to this emerging research area.

5 Conclusion

We presented a well-defined mathematical mission impact assessment, based on a probabilistic approach, without introducing score-based propagation algorithms returning spurious results.

We relied on the expertise of different experts and merged all views without losing information or forcing an expert into a knowledge field he cannot understand. Based on an established mathematical model, we reduced mission impact assessment onto an well-understood problem in computer science. Experimental results demonstrate scalability of the approach such that large-scale network scenarios can be handed.

Future work is dedicated to integrating the presented mission impact assessment into a fully automated cyber-defense system.

Acknowledgments

This work has been partially supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 610416 (PANOPTESEC). The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

References

[1] Albanese, M., Jajodia, S., Jhawar, R., Piuri, V.: Reliable mission deployment in vulnerable distributed systems. In: 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop. pp. 1–8. IEEE (2013)

[2] de Barros Barreto, A., da Costa, P.C.G., Yano, E.T.: Using a Semantic Approach to Cyber Impact Assessment. In: STIDS. pp. 101–108 (2013)

[3] Buckshaw, D.L., Parnell, G.S., Unkenholz, W.L., Parks, D.L., Wallner, J.M., Saydjari, O.S.: Mission Oriented Risk and Design Analysis of Critical Information Systems. Military Operations Research 10(2), 19–38 (2005)

[4] D'Amico, A., Buchanan, L., Goodall, J., Walczak, P.: Mission Impact of Cyber Events: Scenarios and Ontology to Express the Relationships between Cyber Assets, Missions, and Users. In: Fifth International Conference on Information Warfare and Security. pp. 8–9 (2010)

[5] Goodall, J.R., D'Amico, A., Kopylec, J.K.: Camus: Automatically Mapping Cyber Assets to Missions and Users. In: Military Communications Conference. pp. 1–7. IEEE (2009)

[6] Henrion, M.: Practical Issues in Constructing a Bayes' Belief Network. In: Third Conference on Uncertainty in Artificial Intelligence (1987)

[7] Jakobson, G.: Mission Cyber Security Situation Assessment using Impact Dependency Graphs. In: Fourteenth International Conference on Information Fusion. pp. 1–8. IEEE (2011)

[8] Musman, S., Temin, A., Tanner, M., Fox, D., Pridemore, B.: Evaluating the Impact of Cyber Attacks on Missions. In: Fifth International Conference on Information Warfare and Security. pp. 446–456 (2010)

[9] Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, New York, NY, USA, 2nd edn. (2009)

[10] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (2014)