SOCCRATES

SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems

D5.1

Definition of the business model structure, inputs and interfaces

Deliverable type:	Report
Contributing work packages:	WP5 Impact Quantification
Due date of deliverable:	31/08/2020
Submission date:	31/08/2020
Dissemination level:	PU

Responsible organisation:	IMT
Editors:	Christophe Kiennert (IMT)
Revision:	1.0

Abstract	This deliverable describes the Business Logic modelling used for the Business Impact Analyser of the SOCCRATES platform. It describes the model structure, inputs and interfaces.
Keywords:	Security Automation, Business Impact, Model, SOC automation, CSIRT.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 833481
Call H2020-SU-ICT-2018 • Innovation Action • Start date: September 1st, 2019

Classification level: Public

Contributor(s)	Christophe Kiennert (IMT), Fabien Charmet (IMT), Martin Eian (MNO), Maciej Kosz (VTF)

Reviewer(s)	Franck Fransen (TNO)

Security Assessment	**See deliverables table for security assessment requirements**
Approval Date	27/08/2020
Remarks	None

Management summary

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform for SOCs and CSIRTs that will significantly improve an organisation's capability to quickly and effectively detect and respond to new cyber threats and ongoing attacks. To do so, it is necessary to predict the attack path in the infrastructure, which is the role of the Attack Defence Graph analyser, but also to estimate the operational impact on business functions and processes in case the attack succeeds, which is the role of the Business Impact Analyser.

Relying on a Business Logic Modelling based on Bayesian networks, the Business Impact Analyser builds a graph representing the dependencies between the technical assets and the business functions and processes, and performs an analysis of the operational impact of attacks targeting technical assets. To do so, it computes two metrics:

- the probability of impacting a specific business function or process; and
- the critical time, under which no significant operational impact on this business function or process is to be expected

Combined with an estimation of the criticality of each business function and process, this output will allow the SOC analyst to assess the emergency of the situation and adapt the incident response accordingly.

TABLE OF CONTENTS

TABLE OF CONTENTS 4
1 INTRODUCTION – RATIONALE OF THIS DOCUMENT 5
1.1 The SOCCRATES project 5
1.2 This deliverable 6
1.3 Structure of this deliverable 7
2 ROLE OF THE BUSINESS IMPACT ANALYSER 8
3 STRUCTURE OF THE BUSINESS LOGIC MODELLING 11
3.1 Introduction 11
3.2 Bayesian networks 12
3.3 Relevance of Bayesian network formalism in the SOCCRATES project 13
4 DESCRIPTION OF THE MODEL 15
4.1 Definitions 15
4.2 Mission dependency model 16
4.3 Resource dependency model 18
4.4 Shock events 19
4.5 Business impact computation 20
5 INPUTS AND INTERFACES 25
5.1 Mission dependency model inputs 25
5.2 Resource dependency model inputs 27
5.3 Business Impact Analyser query inputs 28
5.4 Interfaces 29
5.5 Future Steps 30
6 REFERENCES 31
7 ABBREVIATIONS 32

1 Introduction – Rationale of this document

This section introduces the SOCCRATES project and defines the goals of this deliverable.

1.1 The SOCCRATES project

SOCCRATES (SOC & CSIRT Response to Attacks & Threats based on attack defence graphs Evaluation Systems) is an EU funded project under the Horizon2020 programme that has the following main challenge:

> ***How can SOC and CSIRT operations effectively improve their capability in detecting and managing response to complex cyber-attacks and emerging threats, in complex and continuously evolving ICT infrastructures while there is a shortage of qualified cybersecurity talent?***

The main objective of SOCCRATES is to develop and implement a security automation and decision support platform (‘**the SOCCRATES platform’)** that will significantly improve an organisation’s capability (usually implemented by a SOC and/or CSIRT) to quickly and effectively detect and respond to new cyber threats and ongoing attacks.

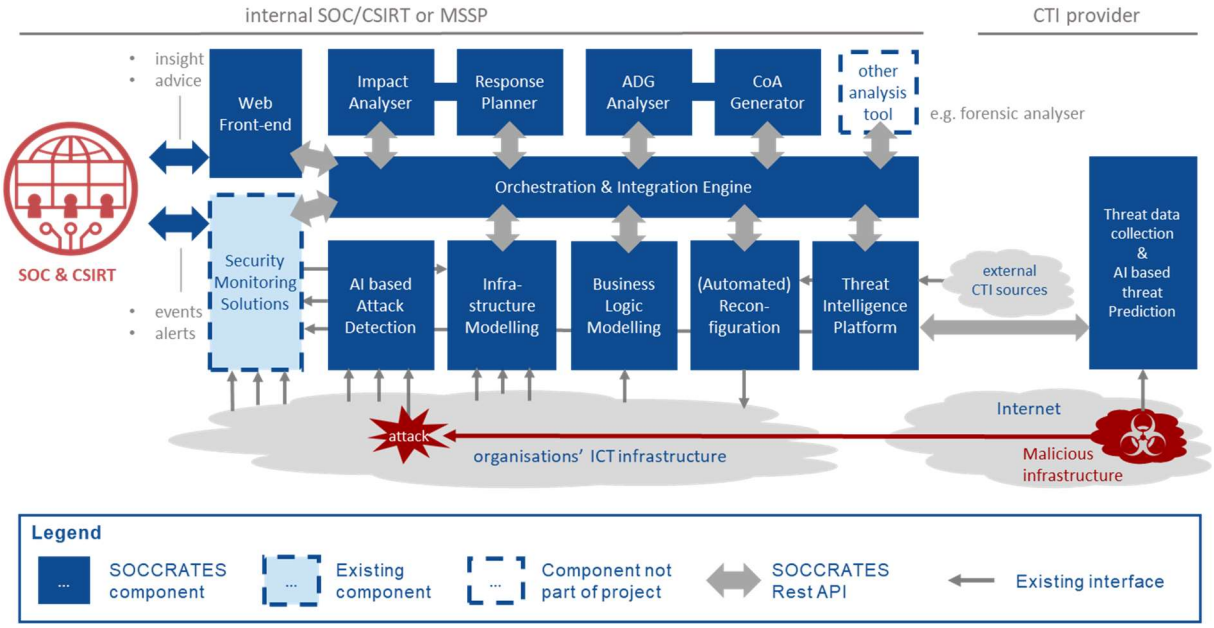

Figure 1.1 – The SOCCRATES platform

The SOCCRATES platform (see Figure 1.1) consists of an orchestrating function and a set of innovative components for automated infrastructure modelling, attack detection, cyber threat intelligence utilization, threat trend prediction, and automated analysis using attack defence graphs and business impact modelling to aid human analysis and decision making on response actions, and enable the execution of defensive actions at machine-speed.

SOCCRATES has the following concrete project objectives:

1. Deliver the SOCCRATES platform consisting of an orchestration function and a unique integration of innovative background solutions that seamlessly work together.
2. Show that the SOCCRATES platform can improve SOC operations by evaluating the SOCCRATES platform in two diverse real-life pilot environments.
3. Examine and illustrate the benefits of automation for selected SOC activities to help manage the cyber security skills gap in organizations.
4. Prepare for successful exploitation by the SOCCRATES partners of the individual innovated components and the integrated SOCCRATES platform in commercial products that are offered to the market and are available for the European (business) community.

Please visit www.soccrates.eu for more information on the SOCCRATES project.

1.2 This deliverable

The purpose of this deliverable is to describe the Business Logic Modelling on which the Impact Business Analyser relies on to perform the business impact analysis on the architecture's assets. The content presented in this deliverable builds on previous work carried out in the PANOPTESEC FP7 project [PANOP], and scientific articles published in peer-reviewed journals [Mot17-1] [Mot17-2], which partly provide the theoretical background for the Business Logic Modelling detailed in this document.

The business impact analysis provides an estimation of the operational consequences of external events such as attacks on the system. These events target one or more technical assets, which might in turn cause business process dysfunction. The probability of impacting the most vital business function in the architecture characterizes the business impact evaluation.

In the SOCCRATES platform, the Business Impact Analyser will be used on assets that were compromised or on which new vulnerabilities were discovered, in order to identify the assets that have the highest business impact value. By doing so, the SOCCRATES platform can prioritize countermeasures on the most vital assets in terms of operational impact.

To perform such evaluation, it is necessary to establish a model of the system that should be comprised of:

- The technical assets
- The business functions and processes
- The dependencies between assets
- The dependencies between the assets and the business processes

Several types of compromise must be taken into account, namely confidentiality, integrity, and availability. This document will therefore describe how an attack might propagate to the business processes, and explain how to produce outputs to evaluate the probability of impact as well as a time tolerance, under which the business process will not be impacted. Finally, the deliverable will also focus on the interface with the Infrastructure Modelling Component, in order to specify how an input from this component can be derived into valuable information for the instantiation of the business model.

1.3 Structure of this deliverable

Section 2 of the document specifies the role of the Business Impact Analyser in the SOCCRATES platform, and its interactions with the SOCCRATES components according to the various use cases described in deliverable D2.1 [D2.1]. Section 3 presents the general structure and concepts of the Business Logic Modelling. Section 4 presents the details of the Business Logic Modelling. Finally, Section 5 describes the Business logic model inputs and interfaces.

2 Role of the Business Impact Analyser

The main roles of the Business Impact Analyser component in the SOCCRATES platform can be summed up as follows:

- perform a reliable, scalable and dynamic operational impact analysis that can be integrated to the various pilots;
- instantiate a practical business logic model of the functions and processes of a company;
- provide a ranking of the most critical assets in the architecture based on the detected attacks and known vulnerabilities; and
- provide the SOC analyst with relevant information that makes it possible to estimate the level of emergency of a threat.

More specifically, the Business Impact analyser aims at analysing how the impact of an attack on a physical asset (Web server, database, etc.) will propagate to business functions and processes and prevent them from working as intended. It is important to note that the objective is not to focus on the propagation of the attack itself, since this task is performed by the ADG Analyser. Rather, the question to which the Business Impact Analyser answers is: *if an attack successfully reached a specific node in the architecture, how likely is it to prevent business functions and processes from working properly, and how much time do we have before such impact becomes effective?*

Therefore, the Business Impact Analyser will produce two metrics:

- the probability of impacting a specific business function or process given as input; and
- the critical time, under which this business function or process will remain safe from any substantial impact.

On top of these two metrics, each business function or process will be associated to a criticality value, characterizing its importance for the proper functioning of the company.

As shown in Figure 1.1, the Business Impact Analyser only interacts with the Orchestration and Integration Engine. As the Business Logic Modelling will use the infrastructure model data produced by the Infrastructure Modelling Component for building business logic model, it is necessary to look into the interface with the Infrastructure Modelling Component, as described in Section 5.

The Business Impact Analyser will first initialize a business logic model based on the current infrastructure model. This instantiated business logic model will be the base for the impact computations, until an update of the model is required. In each of the five use cases described in deliverable D2.1, the Business Impact Analyser receives a request from the Orchestration and Integration Engine at least once. The detailed view on the requests to the Business Impact Analyser in the different use cases can be summed up as follows:

- Use Case 1: Response on Detected Ongoing Attack
    - Analysis stage: Determine if the detected security event can be classified as a security incident

The analysis of the detected event is performed in two steps. First, the objective is to let the SOC operator determine whether the detected event deserves further analysis or not. At this point, the Business Analyser is requested to compute the business impact for every technical asset potentially affected by the detected event.

Then, if the SOC operator decides the event requires further investigation, a set of technical assets with reduced TTC will be determined by the Attack Defence Graph analyser, and the Business Impact analyser will be requested to compute the business impact for each of these assets. The result of this analysis will help the SOC operator to evaluate the level of emergency of the event, in terms of operational impact.

- Containment stage: Determine containment strategy and initiate execution

At this stage, the security event has already been classified as a security incident and the objective is to find the optimal containment strategy. To do so, the Business Impact analyser is requested to compute the business impact on all the compromised assets. Based on this output, the CoA generator will generate appropriate containment strategies, which will then be ranked by the Response Planner.

- Use Case 2: Response on Newly Received Cyber Threat Intelligence

  - Scenario 1: SOCCRATES platform receives new CTI

The treatment of the newly received CTI may lead the Attack Defence Graph analyser to highlight a set of technical assets with reduced TTC due to potential new vulnerabilities. The Business Impact analyser will be requested to compute the business impact for each of these assets. The result of this analysis will help the SOC operator to evaluate the level of emergency of the event, in terms of operational impact.

  - Scenario 2: Delivery of CTI from a.o. malware forensics data

No request to the Business Impact analyser is planned in this scenario.

- Use Case 3: Response on Newly Discovered Vulnerable Assets

Upon a report from a vulnerability scanner indicating newly discovered vulnerable assets in the architecture, the Attack Defence Graph analyser may highlight a set of technical assets with reduced TTC due to the newly discovered vulnerabilities. The Business Impact analyser will be requested to compute the business impact for each of these assets. The result of this analysis will help the SOC operator to evaluate the level of emergency of the event, in terms of operational impact.

- Use Case 4: Response on Discovered System Configuration Change

Upon discovery of a configuration change on one or more assets in the architecture by the Infrastructure Modelling Component, the Attack Defence Graph analyser may highlight a set

of technical assets with reduced TTC due to new vulnerabilities introduced by the configuration change. The Business Impact analyser will be requested to compute the business impact for each of these assets. The result of this analysis will help the SOC operator to evaluate the level of emergency of the event, in terms of operational impact.

- Use Case 5: Response on Deployment of New Systems in Infrastructure

  Upon update of the infrastructure model following the introduction of new assets in the architecture, the Attack Defence Graph analyser may highlight a set of technical assets with reduced TTC due to new vulnerabilities introduced by the new assets. The Business Impact analyser will be requested to compute the business impact for each of these assets. The result of this analysis will help the SOC operator to evaluate the level of emergency of the event, in terms of operational impact.

3 Structure of the Business Logic Modelling

This section describes the general structure of the Business Logic Modelling, as well as the theoretical background necessary to understand the details of the modelling.

3.1 Introduction

The Business Logic Modelling aims at exhibiting how an impact on a technical asset due to an external event (e.g. an attack) propagates into the architecture and which business functions it has a chance to impact, as well as how likely it is to happen. Therefore, the modelling must represent the technical assets on one hand, and the business processes on the other hand. Obviously, it must also exhibit the link between technical assets and business functions.

For that reason, the Business Logic Modelling is split into two sub-models:

- Mission Dependency model
- Resource Dependency model

The Mission Dependency model describes the dependencies between the various business processes, as well as between the business processes and the technical assets directly related to them. This model provides a high-level view of the operational impact without focusing on a fine-grained understanding of underlying interactions between technical assets.

In contrast, the Resource Dependency model describes the dependencies between the technical assets, independently from their impact on the business processes. This model aims at taking into consideration a fine-grained view on the interactions between technical assets, in order to provide a better evaluation on their probability of impacting business processes.

The architecture of the Business Impact Analyser component can be represented as in Figure 3.1.

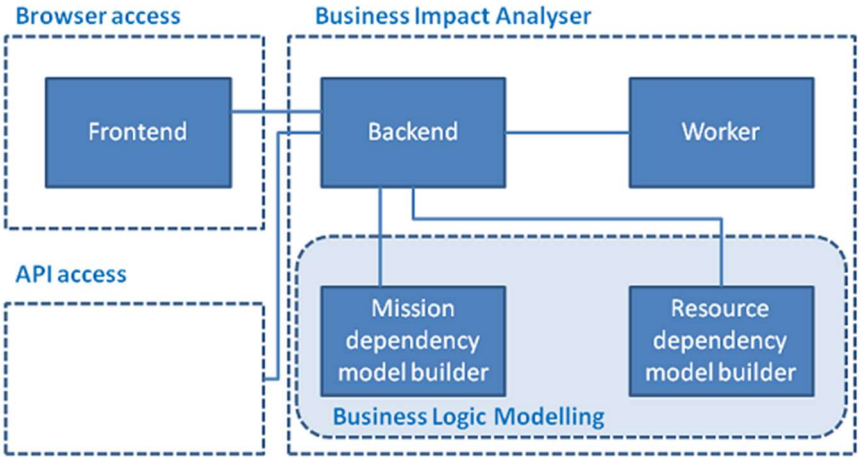

Figure 3.1 – Architecture of the Business Impact Analyser

Both Mission Dependency model and Resource Dependency model are built according to the Bayesian network formalism. Therefore, before describing the Business Logic Modelling in further details, it is necessary to introduce Bayesian networks as a theoretical background.

3.2 Bayesian networks

Bayesian networks are graphical probabilistic models which make it possible to represent local causality and dependencies. Formally, a Bayesian network is a directed acyclic graph where nodes represent the variables and the arcs represent the conditional dependencies between variables. Given a set of observed variables, it is possible to compute the probability distribution of a set of unobserved variables using the conditional dependencies between variables. If the Bayesian network is comprised of *n* variables X_i, then the joint probability distribution is computed as:

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i | Parents(X_i))$$

Figure 3.2 gives an example of a Bayesian network with five nodes A, B, C, D, and E. For example, C can be a binary variable corresponding to whether an alarm is raised in a house, while A can correspond to whether a burglar came into the house, and B to whether an earthquake occurred. Both A and B have a chance to trigger C, which in turn could trigger other events D and E, such as a call from two different neighbors.

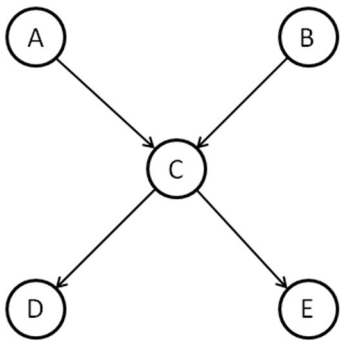

Figure 3.2 – Example of a Bayesian network

In the Bayesian network of Figure 3.2, if we note n^+ and n^- the two possible values of the variable of a node N (with respective meaning true and false), we can for example compute, using the previous formula:

$$P(a^+, b^+, c^-, d^-, e^+) = P(a^+)P(b^+)P(c^-|a^+, b^+)P(d^-|c^-)P(e^+|c^-)$$

For this calculation to be possible, the tables of conditional probabilities must be established beforehand for every node, as well as the prior probabilities for the variables that have no parent (in this example, these are A and B).

Once a Bayesian network has been defined with its graph and conditional probability distributions, it is possible to compute requests on the Bayesian network. Three types of requests can be defined:

- Diagnosis: compute the probability *P(Cause | Consequence)* of a cause given the observation of a consequence
- Prediction: compute the probability *P(Consequence | Cause)* of a consequence given the observation of a cause
- Marginal or joint probability of events

Each of these requests can be computed using the joint probability formula of the network, given above.

For example, in the example above, if we want to calculate the probability that neighbor E will call knowing that a burglar gets into the house, we have to calculate $P(e^+|a^+)$, which can be obtained as follows:

$$P(e^+|\,a^+) = \frac{P(e^+, a^+)}{P(a^+)} = \frac{\sum_{b,c,d} P(e^+, a^+, B = b, C = c, D = d)}{P(a^+)}$$

The various terms of the numerator can each be evaluated using the joint probability formula. Therefore, any request sent to the Bayesian network can be calculated knowing only the conditional probability distributions and the prior probability tables.

3.3 Relevance of Bayesian network formalism in the SOCCRATES project

Since the objective of the Business Impact analyser is to assess the attack impact propagation over business functions with potentially complex dependencies, a probabilistic graphical model expressing causality between nodes, i.e. between various components and functions of the system is the most intuitive and appropriate formalism to rely on. The Bayesian network formalism is one such model, making it possible to efficiently capture and leverage the causality relationships, with well-known results regarding the calculation of event probabilities.

Moreover, Bayesian networks have several advantages that make them particularly relevant for the SOCCRATES project:

- they only require local information about dependencies. In the context of SOCCRATES, that means that the only required expertise to evaluate the weight of the dependencies is limited to that of direct causality relationships, which remain accessible to any person familiar with either the architecture of the company, or with its business aspects. It is therefore not necessary to have an in-depth understanding of the whole architecture and hidden causality relationships to build the model instance;
- they can naturally treat predictive requests, which in the context of SOCCRATES correspond to the request of the probability of impacting a specific business function given the observation of an attack on a technical asset, i.e. the business impact; and
- even though exact probability computation might be intractable depending on the size of the graph, approximate computation can be easily performed through Monte-Carlo simulation, as described in Section 4.

For all these reasons, Bayesian networks appear to be extremely well adapted as the basis for the Business Logic Modelling on which the Business Impact Analyser relies. It is however important to note that Bayesian networks suffer from an important modelling limitation due to the acyclicity assumption of the graph. For example, to capture the dependencies between technical assets where data exchanges can be bilateral, it is natural to represent this situation with mutual dependencies, which lead to cycles in the graphical model. Therefore, even though Bayesian networks provide a very strong basis for Business Logic Modelling, it will be necessary to extend the formalism to non-acyclic graphs.

The next section will therefore detail how Bayesian networks are used to build the Mission Dependency model and Resource Dependency model.

4 Description of the model

This section describes the Business Logic Modelling in detail. This section first provides general definitions, before giving a formal definition and description of the Mission dependency model and Resource dependency model, as well as shock events. Finally, it describes how to compute the business impact on any business function.

4.1 Definitions

This subsection introduces several definitions on which both the Mission dependency model and Resource dependency model rely.

Definition 1 (Impact): *An event is said to have an impact on an asset or a business function if this event causes the asset or the business function to not fully operate as intended. The underlying meaning of "intended operation" depends on the use case.*

Definition 2 (Probabilistic preliminaries): *A node of a probabilistic dependency model is a random variable, denoted with capital letters, e.g., X. A random variable X is assignable to one of its possible values x ∈ dom(X). Let P(X = x) denote the probability of random variable X having x as a value. In the case of a binary variable where dom(X) = {true, false}, we write x^+ for the event X = true and x^- for X = false.*

Definition 3 (From Dependencies to Distributions): *We render every dependency of random variable Y on X as an individual conditional probability $P(x^+|y^+)$ and $P(x^+|y^-)$. Such individual conditional probabilities are fragments of a complete conditional probability distribution (CPD) and are therefore denoted in lowercase.*

To acquire the local CPD P(X|**Y**) of node X from all its individual dependencies on nodes Y ∈ **Y**, we employ combination functions such as noisy-OR, as described by Henrion (in *Practical Issues in Constructing a Bayes' Belief Network*, Henrion, 1988).

The most widely used combination functions are noisy-OR and noisy-AND. The noisy-OR function considers that among all the individual dependencies Y ∈ **Y** on a node X, each single one is sufficient to produce an impact on node X even in the absence of all other dependencies. In contrast, the noisy-AND function considers that all dependencies must be present simultaneously to produce an impact on node X.

In the context of the SOCCRATES project, it appears to be much more relevant to rely on the noisy-OR function, since claiming that a node cannot be impacted if all but one dependencies are applied to the node, as in the case of the noisy-AND function, is a very bold and most of the time unrealistic assumption. In contrast, the noisy-OR function is a safer approach, where the SOC analyst will be given worst case values as output of the business impact analysis.

When the noisy-OR function is applied, the computation of $P(x^+|\mathbf{Y})$ can be done according to the following formula:

$$P(x^+|\boldsymbol{Y}) = 1 - \prod_{y \in \boldsymbol{Y}} (1 - P(x^+|y^+))$$

The purpose of relying on combination functions in a Bayesian network is to reduce the number of inputs to instantiate the model. Indeed, in a Bayesian network, if a node X has *n* binary predecessors $Y_1,\ldots,Y_n$ (hence *n* dependencies), a complete CPD table would require 2^n values as input, since the probability of x^+ must be known conditionally for every combination of the two possible values of each variable Y_i. In comparison, when using a noisy-OR function, only *n* individual conditional probabilities $P(x^+|y^+)$ are necessary.

4.2 Mission dependency model

4.2.1 Graph definition

The Mission dependency model can be formally defined as a graph, as follows.

Definition 4 (Mission dependency model): *A mission dependency model M is a directed acyclic graph (DAG) as a pair* (***V***, ***E***) *of vertices* ***V*** *and edges* ***E****. Vertices* $V \in \boldsymbol{V}$ *are random variables and are categorized according to their semantic as business resources (BR), functions (BF), and processes (BP). For the scope of this work, we consider that a business dependency model is created for a single company, represented by the Business Company (BC) node. The ordering* $BR \prec BF \prec BP \prec BC$ *represents the strict topological ordering of graph M. Every edge* $E \in \boldsymbol{E}$ *represents a dependency.*

Let $V \in \boldsymbol{V}$, then let $\mathbf{E_V} \subseteq \mathbf{E}$ be the set of edges directed to V, and let $\mathbf{D_V}$ be the set of vertices from which $\mathbf{E_V}$ origin, i.e., $\mathbf{D_V}$ is the set of dependencies of V.
For every vertex $V \in \boldsymbol{V}$ a conditional probability distribution (CPD) $P(V|\boldsymbol{D_V})$ is given, or, alternatively, a combination function is given for V and edges $E \in \mathbf{E_V}$ are associated with individual conditional probabilities, i.e. a $p(v^+|d)$ is given for all $d \in dom(D)$, $\forall D \in \mathbf{D_V}$.

It should be noted that depending on the granularity of the Mission dependency model, a number of intermediate categories may be introduced between BF and BP. Typically, a business process may rely on one or several business subprocesses, which in turn might rely on one or several subsubprocesses, etc. There is no theoretical limit to the height of the Mission dependency tree.

An example of Mission dependency model is illustrated in
Figure 4.1. In this model, a business resource A may cause an impact on business functions BF_1 and BF_2, and a business resource B may only cause an impact on business function BF_2. Both business functions BF_1 and BF_2 may in turn cause an impact on business process BP_1, which may in turn cause other impacts on more critical mission nodes.

While being a graphical probabilistic model, this formalism aims to remain close to Business Process Model and Notation (BPMN) models. This allows business experts already familiar with BPMN to easily adapt their business models to this formalism, or even to provide a BPMN model as input to derive the mission nodes and edges of the Mission dependency model. Indeed, a BPMN model can be seen as a business process in the Mission dependency model, while a BPMN task corresponds to a business function.

The hierarchy between the mission nodes results from the acyclic nature of the graph (which naturally leads to a tree representation), and allows the SOC analyst to identify the most vital business processes to query business impact on.

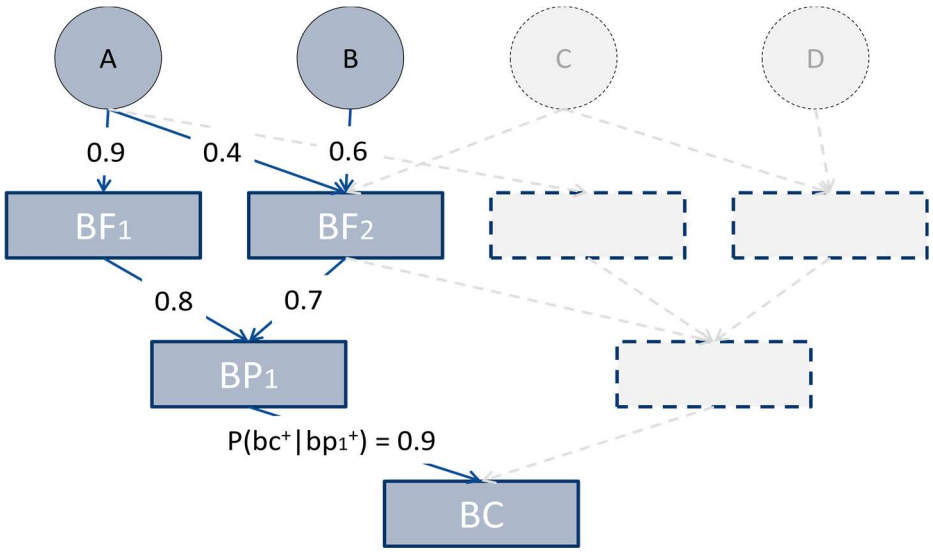

Figure 4.1 – Mission dependency model

The lowest nodes in the hierarchy, called business resources (nodes A and B in Figure 4.1), are actually the technical assets that have the most direct impact on business functions. In other words, business resources provide business functions to some extent. However, these business resources might also have dependencies with other technical assets, which are captured in the Resource dependency model. Therefore, there is no requirement to accurately choose the business resources to include in the Mission dependency model, since the underlying Resource dependency model will adjust how the impact propagates from an attack on a technical asset to a given mission node.

To be instantiated, this model requires a description of all the business functions and processes of the company, as well as their underlying hierarchy. It also requires an evaluation of the value $p(v^+|d)$ of the dependencies. A methodology to derive these elements from raw data will be discussed in Section 5.

It should also be noted that although the model instantiation can require a high level of expertise in order to obtain an accurate description of the infrastructure and of the dependencies between the mission nodes, the model is likely to remain stable for a rather long period of time, with only minor updates that are likely to be scarce. Indeed, even though there can be frequent and substantial changes in the technical assets deployed in the company's infrastructure, it does not necessarily entail any change in the business functions and processes carried out by the company. Therefore, once an initial Mission dependency model has been established, there is no need for the same level of business expertise to keep the model updated over time, unless substantial change happens in the company's activity.

4.2.2 Temporal aspects

Even though the graph of business processes and business functions is likely to remain mostly stable in time, several temporal aspects still have to be taken into account in the model.

First, the evaluation of the conditional probability distributions is likely to depend on:

- the time of the day: during the night, the impact on business functions or processes is likely to be considerably different from the impact during working hours; and

- the time of the year: during a period of intense activity (which, for a commercial company, could correspond to periods like Christmas), the impact on business functions or processes is likely to be considerably different than a period with usually little activity, such as summer vacations

Therefore, several sets of conditional probability distributions should be defined beforehand in order to capture these temporal variations.

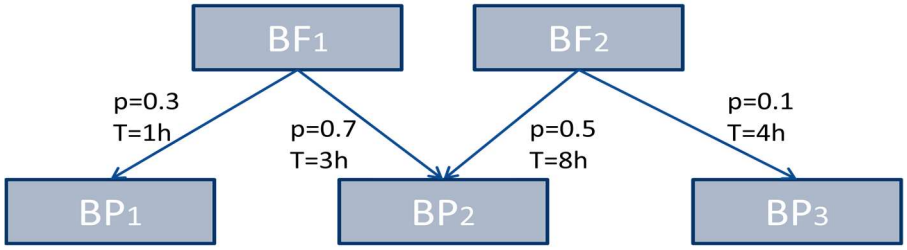

Figure 4.2 – Mission dependency model with downtime tolerances

Second, the impact on a business function will not necessarily propagate to a higher-level business process, provided that the business function is not impacted for too long. This downtime tolerance may depend on the higher-level business processes that might be impacted. Therefore, as shown in Figure 4.2, for each dependency, on top of the conditional probability, a value *T* will be added, corresponding to the downtime tolerance of the business function before propagating the impact to the higher level business process. In turn, if the downtime tolerance is intrinsic to the business function, all the dependencies to higher level business processes will be associated to the same downtime tolerance.

4.3 Resource dependency model

The Resource dependency model can be formally defined as follows.

Definition 5 (Resource dependency model): *A mission dependency model M is a directed graph as a pair* (***V***, ***E***) *of vertices* ***V*** *and edges* ***E****. Vertices V ∈* ***V*** *are random variables and represent technical assets in the infrastructure. Every edge E∈****E*** *represents a dependency.*

Let V ∈ ***V***, then let $\mathbf{E_V} \subseteq \mathbf{E}$ be the set of edges directed to V, and let $\mathbf{D_V}$ be the set of vertices from which $\mathbf{E_V}$ origin, i.e., $\mathbf{D_V}$ is the set of dependencies of V.
For every vertex V ∈ ***V*** a conditional probability distribution (CPD) $P(V|\mathbf{D_V})$ is given, or, alternatively, a combination function is given for V and edges E ∈ $\mathbf{E_V}$ are associated with individual conditional probabilities, i.e. $p(v^+|d)$ is given for all d ∈ dom(D), ∀D ∈ $\mathbf{D_V}$.

An example of Resource dependency model is illustrated in Figure 4.3, in which nodes A, B, C and D can have mutual dependencies. A subset of the nodes, in this case nodes A and B, can be part of the Mission dependency model (cf. nodes A and B in
Figure 4.1), if they represent technical assets that have been identified as having a direct impact on at least one business function. Node SE_B represents an external shock event occurring on node B. Shock events will be described in Section 4.4.

Even though the definition of the Resource dependency model is close to that of the Mission dependency model, an important difference is that cycles are allowed in the Resource dependency model. Indeed, the assumption of an acyclic graph cannot hold in the case of the dependencies be-

tween technical assets, as the data exchanges cannot be supposed to be unilateral. Therefore, the Resource dependency model is not a Bayesian network, even though the impact on a node is propagated in a similar way as in a Bayesian network.

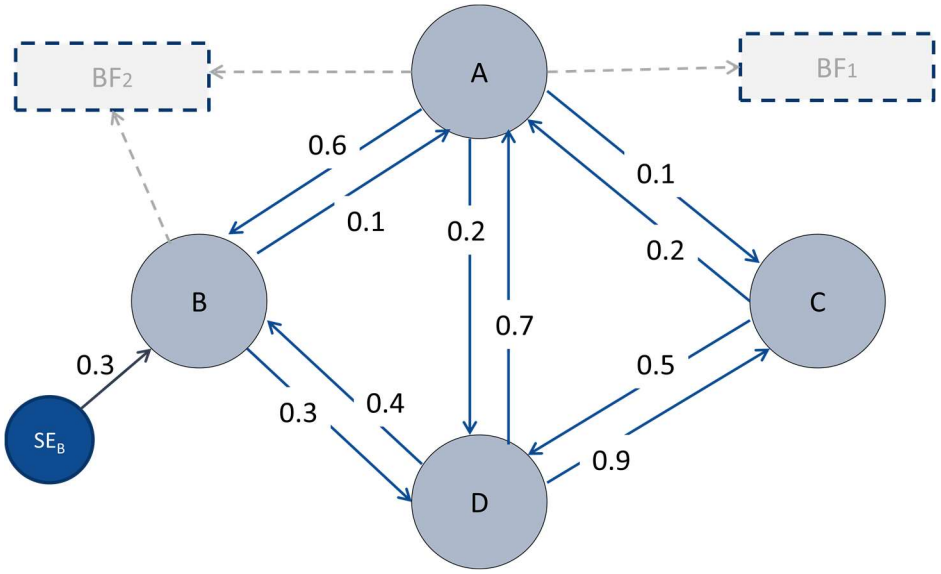

Figure 4.3 – Resource Dependency Model

The objective of the Resource dependency model is to focus on the assessment of the dependencies between technical assets independently from their functions in the company, since such knowledge may not be in the scope of an expert establishing the company's Mission dependency model. The technical assets can be any kind of physical or virtual component deployed in the company's infrastructure (server, device, software, connected objects, etc.).

Similarly, to the Mission dependency model, this model requires an evaluation of the value of each dependency, seen as a local probability of impact. However, contrary to the Mission dependency model, the Resource dependency model is likely to evolve on a regular basis, either with the addition of new assets, the removal of assets, or configuration changes that may modify the dependencies. Therefore, such evaluation cannot be performed by hand and must be automated. An approach for the evaluation of the various dependencies, basic on the measured traffic exchanges in the architecture, will be described in Section 5.

4.4 Shock events

In Bayesian networks, the input values of a model are the local conditional probability distributions for each node, or alternatively, as stated in the previous sections, local individual conditional probabilities associated with a combination function, such as a noisy-OR function. However, the nodes that do not have any predecessors are the only ones for which non-conditional probability distributions must be provided. In the context of SOCCRATES, since the probability of being impacted is supposed to be dependent of an external event (typically an attack or a CoA), these source events are represented in the Resource dependency model as extra nodes, called shock events. In the example of Figure 4.3, there is one shock event SE_B affecting node B.

Formally, an external shock event can be defined as follows.

Definition 6 (External Shock Event): *An external shock event SE is a random variable and let* ***SE*** *be the set of all known external shock events. An external shock event SE ∈* ***SE*** *might be present (se^+) or not be present (se^-), according to a prior random distribution P(SE). Every vertex V of a resource dependency model might be affected by one or more external shock events* $\mathbf{SE_V} \subseteq \mathbf{SE}$*. In the case that an external shock event is present (SE = se^+,* $SE \in \mathbf{SE_V}$*), there exists a probability of it affecting node V, expressed as a local conditional probability fragment $p(v^+|se^+)$.*

Since a local impact on a node V can only exist when a shock event is present, we always have $p(v^+|se^-) = 0$.

Every individual conditional probability fragment from an external shock event can be treated by a combination function such as noisy-OR, in the case where multiple shock events can affect one node, similarly to what was defined for the Mission and Resource dependency models.

The evaluation of the probability of impact of a shock event cannot be done in the same way as all other conditional probability distributions, which are related to the data exchanged between assets. In the case of a shock event, the probability of impact is related to the level of compromise of the attack, and to the level of sensitivity of the asset. This issue is detailed in Section 5.3.

Another aspect of shock events is that they may have a different impact on a given resource node over time. For instance, if the external shock event is a patch applied on the resource node, then it will first lead to a temporary shutdown of the node, while afterwards it will have no further negative impact on the node availability. Therefore, the conditional probability in terms of impact on availability would be 1 at first, and then be down to 0. To take such cases into account, the temporal aspect of external shock events can be defined as follows.

Definition 7 (Temporal Aspect): *The temporal aspect of a shock event can be captured by defining abstract timeslices, in which the prior and local conditional probabilities may vary. Given T+1 timeslices $t_0,\ldots t_T$, the prior and local conditional probabilities for a shock event must be defined as a T+1 tuple $\langle t_0 : p_0,\ldots, t_T : p_T \rangle$ with one value per timeslice.*

This temporal aspect makes it possible to provide a finer grain analysis of the impact of external shock events. However, in the context of the SOCCRATES project, the temporal aspects will be mainly taken into consideration in the mission dependency model, as detailed in Section 4.2.2. Although the formalism used for the Business Logic Modelling makes it possible to add a temporal dimension for shock events, it requires a very precise understanding of the dynamics of the attacks targeting the system, which is not adapted to a real time analysis.

4.5 Business impact computation

In this section, we describe how to compute the probability when a shock event may impact a business resource, or a node from the Resource dependency model. We will first detail the theoretical computation of the impact probability, and then we describe a Monte-Carlo approximation to make the problem computationally tractable.

It should be emphasized that the Business Impact Analyser does not compute a single “impact” value, but two metrics: impact probability, and critical time. Based on these two metrics, and on a criti-

cality value associated to each business function and business process that will help assess the severity of the attack impact for the company, the SOC analyser will be able to assess the level of emergency of the situation.

4.5.1 Impact probability

4.5.1.1 Theoretical approach

We note ***SE*** the set of all known shock events, V a vertex of the mission dependency model, and $\boldsymbol{SE}_V \subseteq \boldsymbol{SE}$, the set of shock events impacting node V**.**

We note $\boldsymbol{C}_{SE-V}$ the set of paths from shock event SE to vertex V.

We note $\boldsymbol{C}_V$ the union, for every possible SE, of the corresponding $\boldsymbol{C}_{SE-V}$**:**

$$\boldsymbol{C}_V = \bigcup_{SE \in \boldsymbol{SE}} \boldsymbol{C}_{SE-V}$$

Each path c ∈ $\mathbf{C}_V$ is the set of vertices connecting a shock event to node V.

It should be noted that $\mathbf{C}_V$ might contain redundant paths. For example, in the business logic model from Figure 4.5, the path $SE_B \rightarrow B \rightarrow A \rightarrow BF_1$ is redundant with the path $SE_B \rightarrow B \rightarrow C \rightarrow B \rightarrow A \rightarrow BF_1$, since both paths contain the edges (SE_B, B), (B, A), and (A, BF_1). If the impact successfully propagates from SE_B to B, from B to A, and then from A to BF_1, then it is irrelevant, when computing the impact propagation to node BF_1, to consider any other path that contain these two edges. Therefore, we henceforth consider that $\mathbf{C}_V$ only contains the shortest of each set of redundant paths.

Definition 8 (Impact probability): Under the noisy-OR assumption, we define the impact probability on a node V as the probability of the set of shock events ***SE*** causing an impact on node V.

We note this probability $p(V^+|\boldsymbol{SE})$, and it is computed by considering each path in $\mathbf{C}_V$ as a conjunction of Boolean random variables corresponding to the edges of the path (with probability distributions given by the conditional probability distributions defined in Sections 4.2 and 4.3), and then computing the disjunction of all conjunctions in $\mathbf{C}_V$. Indeed, under the noisy-OR assumption, only one path needs to have all its edges at the value *true* to propagate the impact from ***SE*** to V.

Therefore, the probability $p(V^+|\boldsymbol{SE})$ can be computed as follows, where **P**(c) represents the probability of path c viewed as a set:

$$p(V^+|\boldsymbol{SE}) = \bigcup_{c \in \boldsymbol{C}_V} \boldsymbol{P}(c)$$

However, as illustrated in Figure 4.4, the probabilities of the different paths are not independent from each other. Therefore, the overlapping edges of these different paths must not be counted multiple times when determining the probability $p(V^+|\boldsymbol{SE})$.

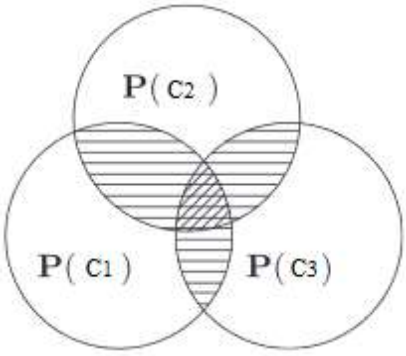

Figure 4.4 – Overlapping probabilities between paths in C_v

In order to count each part of the different sets only once, we use the inclusion-exclusion principle.

The inclusion-exclusion principle states that the union of *n* sets is computed by adding the odd-tuple-wise intersections of sets and subtracting the even-tuple-wise intersections of sets.

In other words, the union is the sum of all intersections of an odd number of sets to which is subtracted the sum of all intersections of an even number of sets.

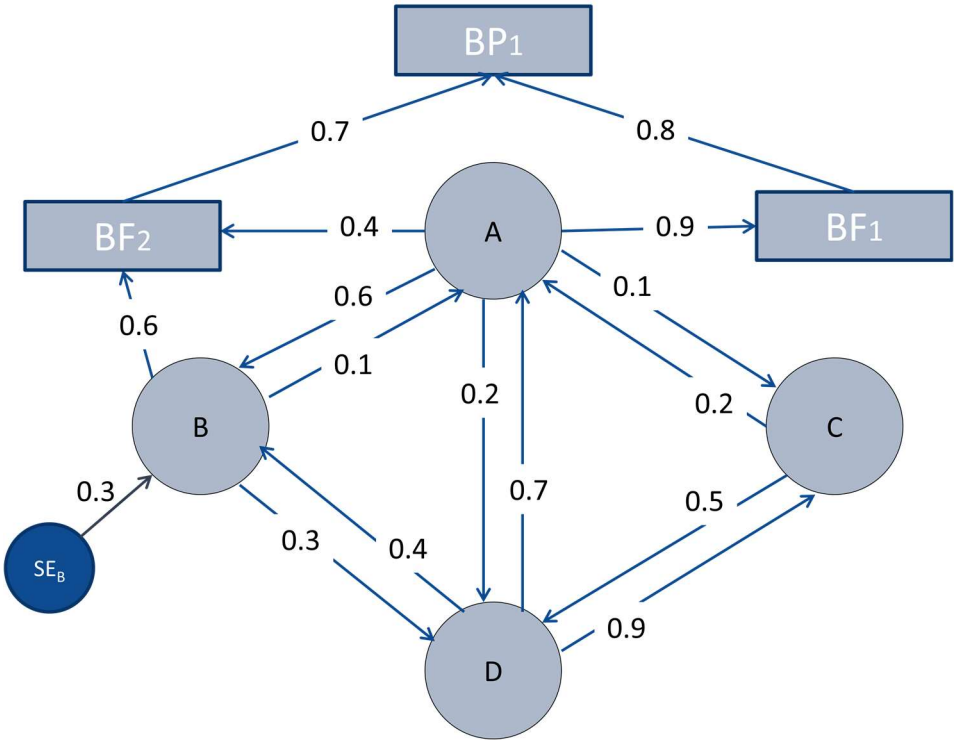

Figure 4.5 - Business Logic model with conditional impact probabilities

We consider the business logic model depicted in Figure 4.5 as an example. This model is a subset of the union of the Mission Dependency model from Figure 4.1 and of the Resource Dependency model from Figure 4.3.

Let us consider the case where we want to compute the probability of business function BF_1 being impacted by the shock event SE_B occurring on node B, i.e. $p(bf_1^+ | SE_B^+)$.

There are three different paths from shock event SE_B to node BF_1:

- $c_1 = SE_B \rightarrow B \rightarrow A \rightarrow BF_1$
- $c_2 = SE_B \rightarrow B \rightarrow D \rightarrow A \rightarrow BF_1$
- $c_3 = SE_B \rightarrow B \rightarrow D \rightarrow C \rightarrow A \rightarrow BF_1$

For each of these paths, the probability can be computed as follows:

$$p(c_1) = p(bf_1^+|a^+)p(a^+|b^+)p(b^+|SE_B^+)$$
$$p(c_2) = p(bf_1^+|a^+)p(a^+|d^+)p(d^+|b^+)p(b^+|SE_B^+)$$
$$p(c_3) = p(bf_1^+|a^+)p(a^+|c^+)p(c^+|d^+)p(d^+|b^+)p(b^+|SE_B^+)$$

According to the inclusion-exclusion principle:

$$p(bf_1^+|SE_B^+) = p(c_1) + p(c_2) + p(c_3) - p(c_1, c_2) - p(c_1, c_3) - p(c_2, c_3) + p(c_1, c_2, c_3)$$

The probability $p(c_1,c_2)$ is computed as the product of the probabilities associated to each edge appearing at least once in c_1 or in c_2.

$$p(c_1, c_2) = p(bf_1^+|a^+)p(a^+|b^+)p(a^+|d^+)p(d^+|b^+)p(b^+|SE_B^+)$$

Similarly, $p(c_1,c_2,c_3)$ is computed as:

$$p(c_1, c_2, c_3) = p(c_1, c_2)\, p(c^+|d^+)$$

4.5.1.2 Monte-Carlo Approximation

Since the exact computation time of the inclusion-exclusion principle grows exponentially with the number of nodes in the resource dependency graph, the impact probability can be computed with a Monte-Carlo approximation.

The procedure can be described as follows:

1. For each business resource node, use a Depth First Search algorithm to find all possible paths from every shock event node.
2. For each path, instantiate every random variable of the path (give them values **true** or **false**), according to their probability distribution.
3. The conjunction of random variables is then evaluated to determine if the business resource is impacted. If it is, there is no need to evaluate the other paths, due to the Noisy-OR hypothesis.
4. The business resource node is ticked, and a counter associated to the business resource is incremented.

We repeat steps 1-4 for all business resources, while accounting for Noisy-OR and Noisy-AND hypotheses. This implies that a Noisy-AND hypothesis forces to verify all elements of the disjunction (as opposed to a conjunction where only one must be verified).

The procedure is performed a *n* times, and the probability $p(br^+)$ for a business resource to be impacted is evaluated as follows:

$$p(br^+) = \frac{counter_{br}}{n}$$

4.5.2 Critical time

The critical time corresponds to the time under which some impact to a business process or function will not cause any significant operational impact for the company. For example, if a business resource is targeted by an attack during the night, a few business functions supported by this resource might not work properly without causing any operational impact to the company, since such business functions might not be required during night-time. In particular, the impact will not immediately propagate to business processes which rely on these business functions.

A downtime tolerance is therefore associated to each dependency in the Mission dependency model, on top of the conditional probability. Figure 4.6 illustrates the same business logic model as Figure 4.5, but displaying the downtime tolerances (e.g. in hours) instead of the conditional probabilities. Since this downtime tolerance has no relevance for the resource dependency model, the dependencies between assets do not need to have an associated downtime tolerance.

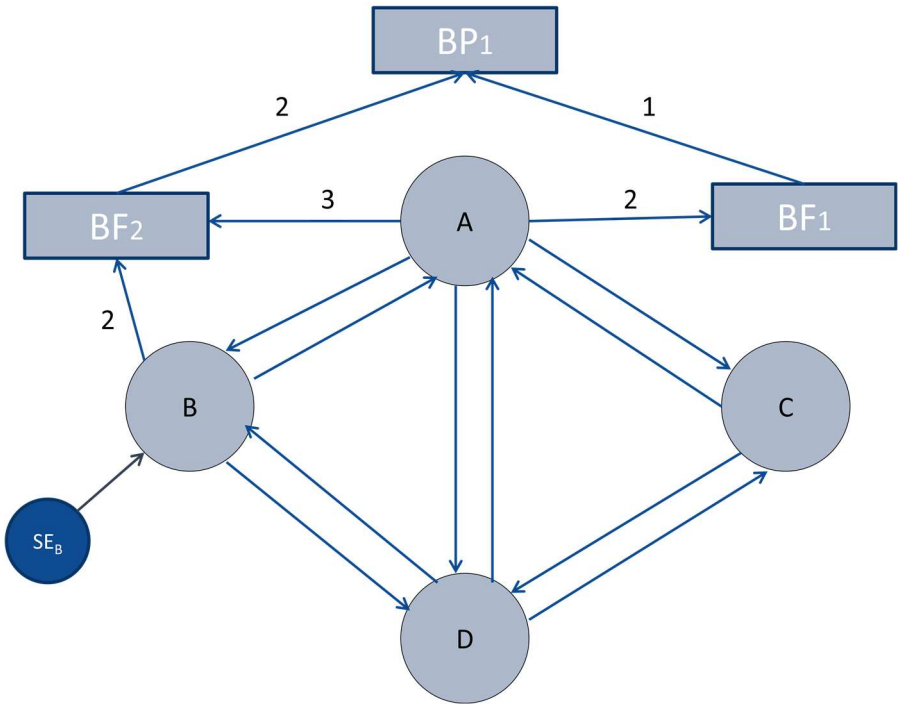

Figure 4.6 – Business Logic model with downtime tolerances (in hours)

The computation of the critical time is performed according to the research of the shortest path from the shock event to the business function or process on which the impact analysis is performed. This can be done with a classic Shortest Path algorithm such as the Dijkstra algorithm, taking a value of 0 for the dependencies between assets that are not relevant in terms of downtime tolerance.

In the example of Figure 4.6, the critical time for business process BP_1 would be 3 hours, corresponding to the path $SE_B \rightarrow B \rightarrow A \rightarrow BF_1 \rightarrow BP_1$.

5 Inputs and interfaces

This section aims at describing how the Business Impact analyser component interacts with the other components of the SOCCRATES architecture, and more specifically, the inputs required to instantiate the full business model of the company, before conducting business impact analyses.

5.1 Mission dependency model inputs

In order to build the mission dependency model, the following inputs are required:

- List of business processes and business functions
- List of business resources supporting business functions
- Dependencies (conditional probability distributions)
- Downtime tolerance for each business process and business function
- Criticality of each business process and business function for the organisation

Building the mission dependency model is a process that cannot be fully automated. For that reason, this model aims at remaining as stable in time as possible, so that manual tasks are mainly kept to the initialization of the model. However, partial automation can be achieved regarding the list of business processes, functions and resources.

Elements of methodology to obtain the various inputs of the mission dependency model are described below.

5.1.1 List of business processes, functions and resources

Obtaining the list of business processes and business functions can be partially automated, with various data sources. In particular:

- BPMN models
- Active Directory (AD) tree
- List of company assets

BPMN models correspond to the most reliable data source to build the list of business processes and business functions, since BPMN models specify business processes in a company. Therefore, a BPMN model corresponds to a business process in the mission dependency model, while BPMN tasks correspond to business functions, and subprocesses have a direct equivalence in the mission dependency model. Moreover, the data stores which business tasks access to retrieve data make it possible to identify business resources that support business functions. Figure 5.1 illustrates the link between a BPMN model and the mission dependency model.

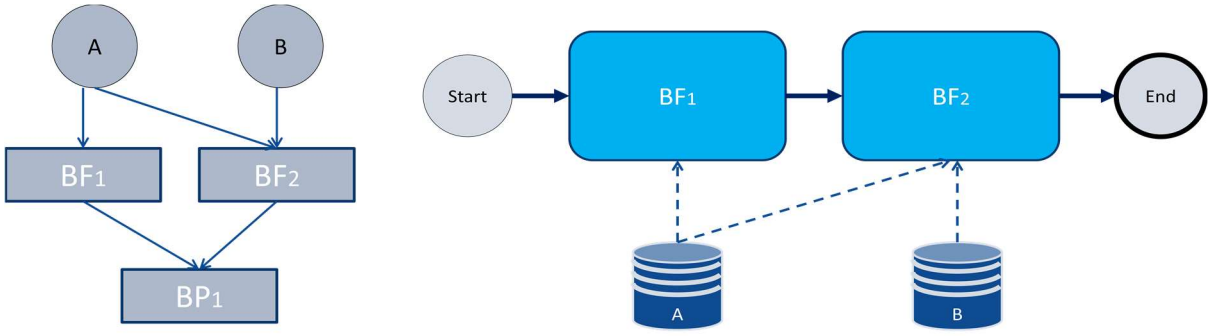

Figure 5.1 – Equivalence for business process BP1 in the mission dependency model (left) and as a BPMN model (right)

The main issue with BPMN models is that a lot of companies do not have such models of their business activity. In contrast, the Active Directory (AD) tree is a data source that can be found in the vast majority of companies, although it is not as reliable as BPMN models to derive a list of business functions and processes. The idea of using the AD tree is to focus on the Organizational Unit (OU) field, which usually contains the various departments of a company. Since it can be expected from a well-organized company to have each department dedicated to at least one business process, this can be used as a coarse-grained approach to find out a relevant list of business processes in the company. The main drawbacks of this data source are its inability to provide business functions, and the implicit assumption that the business processes would exactly follow the company departments, which might be a rough approximation of the reality. Therefore, manual work will be required to refine the results obtained from the AD tree.

Finally, heuristic approaches led on the physical assets of the company infrastructure can facilitate the identification of business resources and business functions, by analysing which of the assets are directly essential for operational activities, and which are more dedicated to support functions. For example, most servers are likely to support business functions, while in most companies, assets like firewalls are likely to be dedicated to increase the security of the architecture without supporting any business function themselves.

Alternatively, The Configuration Management Database (CMDB) can be used to link business resources to applications and to business functions or processes. Such data source can facilitate the research of business resources once a list of business functions has been established. At this point, there is still no existing way to automate this research, but this will be investigated later during the project.

On top of these data sources, interviewing a company worker having a good understanding of the business activities makes it possible to refine the results obtained from the previous data sources. The following questions can help figuring out the list of business processes (BP), functions (BF) and resources (BR):

- (BP) What are the main activities of the company, as presented to customers or partners?
- (BP, BF) If this business function/subprocess was down, how would it impact the operational activities of the company?
- (BF) Which activity, if it was completely down, could cause this business process/subprocess not to work properly?
- (BR) Which assets, if down, have a direct impact on the production of at least a department of the company?

- (BF) For the identified assets from the previous question, what are their main operational functions for the company?

5.1.2 Dependencies, downtime tolerance, and criticality

The evaluation of the dependencies between business resources, business functions and business processes, as well as the downtime tolerance, will require expert knowledge on the business activities of the company.

To evaluate the conditional probability distributions, only local knowledge is required. In other words, to evaluate the dependency between a given business function and a given business process, it is sufficient to focus on the direct impact on that specific business process without taking into account other business functions or processes. To facilitate the evaluation of this dependency, it can be useful to question in which proportion a total dysfunction of the business function will affect the business process. Moreover, as mentioned in Section 4, it is relevant to evaluate this dependency in various situations, depending on the time of occurrence, whether during day or night, whether in a period of expected intense or calm activity, etc., in order to predefine a number of static models.

The downtime tolerance is an estimation of how long the business functions or processes can be impacted without propagating the operational impact to other business processes. For example, if the Web server of the company was targeted by an attack in integrity or availability during night time, at least several hours of downtime tolerance can be estimated before it impacts a business process of the company (such as dissemination), whereas during daytime, a much shorter downtime tolerance is to be expected.

Finally, the criticality of business functions and processes will be evaluated as a number between 0 and 10 in order to assess the severity of the impact in the case it reaches the business function or process for which the analysis is performed. To do so, it is necessary to have a good understanding of the business activities of the company in order to estimate to which extent a failure on a specific business function or process will disrupt the company activities.

5.2 Resource dependency model inputs

In order to build the resource dependency model, the following inputs are required:

- List of assets
- Assets sensitivity in confidentiality (C), integrity (I), availability (A)
- Traffic matrix

As pointed out in Section 4, the resource dependency model will be frequently updated due to the evolution of the architecture assets and traffic dynamics. Therefore, full automation to build the model is required.

The list of assets will be obtained from the Infrastructure Modelling Component, and may also provide sensitivity values for asset in terms of confidentiality, integrity, and availability. For example, a high value in confidentiality on an asset means that this asset contains highly confidential data. Note that these figures typically depend on a human operator to enter these values into a software application. In case of the IMC, this is one of the tools that provide source data to the IMC, such as vul-

nerability scanner RADAR from F-Secure. The values are usually based on a Business Impact Assessment (BIA)[1].

The dependencies between the assets will be derived from a traffic matrix quantifying the amount of data exchanged between each pair of assets. The amounts of exchanged data can be obtained from flow analysis features of the NetFlow functionality from Cisco routers. The choice to rely on such traffic matrix to evaluate the dependencies between assets is justified by three main reasons. First, it is a data source that can be reasonably expected from a company, as opposed to finer-grained data at software layer that are much harder to access and exploit. Second, this data source meets the requirement of a fully automated model building, with directly exploitable data. Third, when data are unlawfully modified on an asset, or the asset is made unavailable, it is natural to observe the data flows to estimate how the impact of the attack will propagate in the architecture.

Based on this traffic matrix, if several assets $Y_1,...,Y_n$ send data to a same asset X, and if $T_{Y_i,X}$ represents the amount of traffic sent from Y_i to X, the dependencies will be computed as follows:

$$p(x^+|y_i^+) = \frac{T_{Y_i,X}}{\sum_j T_{Y_j,X}}$$

In other words, the probability that an impact on asset Y_i propagates to asset X is proportional to the amount of data sent from Y_i to X.

5.3 Business Impact Analyser query inputs

Once the current model instance is built based on the previous inputs, the Business Impact Analyser will be able to receive queries to compute the probability of impact on a given business function/process, as well as the time tolerance, given an attack targeting an asset in the architecture (therefore acting as a shock event).

To do so, the following inputs are required:

- Asset targeted by the attack
- Type of attack and level of compromise (in terms of C, I and A)
- Timestamp
- (Optional) Business function or process for which the output should be computed

The business impact analysis requires at least the description of one shock event, which in this case is an attack. Even though the traffic matrix makes it possible to evaluate all the dependencies between assets, the probability that the shock event has an impact on the targeted node is not part of what can be evaluated with the traffic matrix. Therefore, this probability must be evaluated according to the severity of the attack, taking into account both the level of compromise and the sensitivity of the asset. An attack with a strong level of compromise in integrity targeting an asset with a high integrity sensitivity will have a very high probability of impact, while the same attack on an asset with a low integrity sensitivity (i.e. with data not particularly sensitive to unlawful tampering) will have a lower probability of impact on that node. Moreover, attacks on data confidentiality (including personal data) are likely to have potentially high financial impact (taken into account in the evaluation of

[1] Such as the business impact assessment (phase B) of IRAM2 from the Information Security Forum; https://www.securityforum.org/uploads/2015/03/ISF-IRAM2-ES.pdf

courses of actions by the Response Planner), but insignificant operational impact, since a leak of confidential data will usually not prevent any business function of process from working as intended.

Therefore, given an attack representing a shock event *SE* targeting asset *N*, if a_i and a_a represent the level of compromise respectively in integrity and availability evaluated between 0 and 1, while s_i and s_a represent the asset sensitivity respectively in integrity and availability evaluated between 0 and 1, then the probability of impact on that asset can be computed as:

$$P(n^+|SE^+) = \min(a_i s_i + a_a s_a, 1)$$

This formula takes into account both the integrity and availability dimensions to evaluate the severity of the impact. If either the level of compromise or the sensitivity of one dimension is equal to 0, then this dimension is ignored, as could be expected. Moreover, compared to the average value $\frac{a_i s_i + a_a s_a}{2}$, this formula makes it possible to consider the case of attacks focused on only one dimension. For exemple, if $a_i = s_i = 1$, and $a_a = s_a = 0$, the proposed formula gives a probability of impact equal to 1, which is to be expected, while the average value would give a probability equal to 0,5. The drawback of this formula might be a slight overestimation of the probability of impact, which will not be an issue in the context of assessing the impact of a security event.

Since the impact analysis is conducted in the hypothesis that the attack successfully compromised the targeted node, the prior probability distribution of the shock event can be set to $P(SE^+) = 1$.

The timestamp will be used to load the correct mission dependency model to evaluate the business impact, since the values of the dependencies and downtime tolerances are time dependent. Then, the critical time can be computed with a shortest path algorithm such as the Dijkstra algorithm, as detailed in Section 4.5.2.

If no business function or process is specified in the request, then the most vital mission node, which is the highest in the graph tree, will be selected for the impact analysis.

5.4 Interfaces

The Business Impact Analyser has three interfaces with external components:

- Interface with the Infrastructure Modelling Component
- Interface with the ADG analyser
- Interface with the Orchestrator & Integration Engine

The interface with the Infrastructure Modelling Component aims at receiving the data to build or update the resource dependency model. More specifically, the following data will be received and processed:

- List of assets
- NetFlow data
- Asset sensitivities

These three elements will make it possible to compute the graph of dependencies between assets, while relying on the NetFlow data to build the traffic matrix from which the conditional probability distributions will be deduced.

The interface with the ADG analyser aims at receiving the input of the attack for which it is necessary to perform a business impact analysis. The ADG analyser will provide the type of attack and level of compromise (in terms of C, I and A).

The interface with the Orchestrator & Integration Engine aims at ensuring that the Business Impact analyser interacts with the other components of the SOCCRATES platform in compliance with the scenarios described in the different use cases. In particular, the Orchestrator & Integration Engine will transmit queries for business impact analysis to the Business Impact Analyser, and dispatch the output to the correct component.

A third interface might be added in the future with the Response Planner, in order to integrate the business impact of the courses of actions in the evaluation of their financial impact.

5.5 Future Steps

The main purpose of the Business Impact Analyser at the current time is to assess the operational impact of an attack targeting an asset of the infrastructure. However, the shock event nodes described in the formalism (cf. Section 4.4) do not necessarily correspond to attacks. They can represent any kind of external event that may have an impact on an asset of the infrastructure. Thus, it is also possible to represent the CoAs as shock events in this formalism, and therefore to assess the operational impact of CoAs. Such evolution of the Business Impact Analyser will be considered later in the project in order to make it possible for the Response Planner to capture the operational impact of CoAs when evaluating their financial impact for the company.

In a more long-term view, proper research work could be conducted on the automation business functions and processes discovery. Indeed, the main practical limitation of the Business Impact Analyser is the manual work required to establish the Mission Dependency model, especially in the case where no BPMN models of the business processes exist in the company. Developing such automated discovery approach would greatly enhance the applicability of the Business Impact Analyser in a large variety of companies, including those without the required skills to construct a consistent representation of their business functions and processes.

6 References

[Mot17-1] A. Motzek et al., “Selection of Pareto-efficient response plans based on financial and operational assessments”, *EURASIP Journal on Information Security*, 2017 (12)

[Mot17-2] A. Motzek and R. Möller, “Context- and bias-free probabilistic mission impact assessment”, *ELSEVIER Computers & Security 65*, 2017

[PANOP] PANOPTESEC project, FP7-ICT-2013-10 Objective

[D2.1] Martin Eian, S. T. (2019). D2.1 Use cases definition & pilot sites requirements. SOCCRATES. Available via: https://www.soccrates.eu/wp-content/uploads/2019/12/SOCCRATES_D2.1-20191202-FINAL.pdf

7 Abbreviations

Acronym	Description
AD	Active Directory
ADG	Attack Defence Graph
BF	Business Function
BP	Business Process
BPMN	Business Process Model and Notation
BR	Business Resource
CERT	Computer Emergency Response Team
CMDB	Configuration Management Database
CoA	Course of Action
CPD	Conditional Probability Distribution
CSIRT	Computer Security Incident Response Team
CTI	Cyber Threat Intelligence
ICT	Information and Communication Technology
IMC	Infrastructure Modelling Component
LDAP	Lightweight Directory Access Protocol
OU	Organizational Unit
SE	Shock Event
SOC	Security Operation Centre
SOCCRATES	SOC & CSIRT Response to Attacks & Threats based on attack defence graph Evaluation Systems
TTC	Time To Compromise